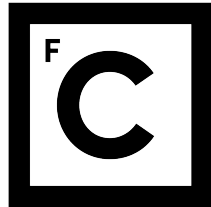


UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Ciências
ULisboa

**A KNOWLEDGE-BASED, SECURE AND
DEPENDABLE SELF-HEALING ARCHITECTURE
FOR THE SMART GRID**

Nuno Emanuel Nunes Pereira

MESTRADO EM SEGURANÇA INFORMÁTICA

Dissertação orientada por:

Prof. Doutor Nuno Fuentecilla Maia Ferreira Neves

2016

Agradecimentos

A entrega deste documento simboliza o final de dois anos em que aprendi muito sobre segurança e sobre mim próprio, mas que ficam marcados pelo tempo que não dediquei à minha família e aos meus amigos.

Agradeço à minha Marijke por todos os dias em que me motivou para continuar a estudar, estando eu cansado e com sono depois de um longo dia de trabalho, e por todas as noites em que cuidou sozinha dos nossos filhos, com todo o amor e carinho, para que eu pudesse ter uma noite de sono descansada e voltar bem ao trabalho e ao estudo no dia seguinte.

Agradeço aos meus filhos João e Teresa por toda a alegria, sorrisos e brincadeiras que contribuíram para que me mantivesse focado em fazer o trabalho bem e depressa para que eu pudesse juntar-me a eles para brincar e aprender com eles.

Agradeço a toda a minha família e amigos pela paciência com todos os almoços, jantares e encontros adiados nos fins-de-semana em que tive de ir às aulas, estudar e/ou trabalhar.

Agradeço ao Professor Nuno Neves por toda a orientação e apoio prestados antes e durante a realização deste trabalho.

Agradeço aos colegas do MSI toda a camaradagem no estudo e na realização dos trabalhos das disciplinas.

Agradeço aos colegas da EDP Distribuição por aturarem as minhas ideias e por todo o apoio prestado no acompanhamento de projectos e iniciativas sempre que eu não pude estar presente.

Obrigado a todos os que percorreram comigo este caminho nos bons e nos maus momentos!

Para a minha família e amigos...

Resumo

As redes de distribuição de eletricidade são infraestruturas críticas que, em casos de incapacitação ou destruição, provocariam um efeito debilitante na economia e na segurança pública. Estas redes são cada vez mais suportadas por sistemas complexos e redes de comunicações, ganhando desta forma alguma inteligência e autonomia. A informação que estes sistemas geram e as decisões e ações que tomam são limitadas pela informação que têm. Em casos nos quais não tenham, por desenho, toda a informação relevante para o seu contexto de atuação, podem enganar os operadores e tomar ações prejudiciais. A dependência dos sistemas e comunicações levanta também preocupações sobre o desempenho, privacidade, segurança e confiabilidade, que se estendem além de possíveis faltas na rede elétrica. Neste sentido, existem soluções dedicadas ao tratamento automático de faltas na rede elétrica, existindo também soluções dedicadas ao tratamento de faltas nos sistemas e comunicações, fazendo-o separadamente. No entanto, como demonstrado pelos incidentes na Ucrânia, no final de 2015, faltas e falhas em diferentes camadas da rede inteligente podem estar relacionadas. Adicionalmente, embora exista alguma preocupação com a segurança e a confiabilidade das soluções de tratamento automático de faltas na rede elétrica no âmbito de alguns projetos europeus, os projetos piloto focam-se maioritariamente nos aspetos funcionais destas soluções, o que poderá comprometer a segurança de futuras instalações.

Em resposta aos problemas acima descritos, nesta tese utiliza-se uma abordagem com base em conhecimento e segurança para desenhar e propor um sistema de tratamento automático de faltas na rede inteligente, que explora as ligações atrás mencionadas. Inicialmente, são definidos requisitos de alto nível para as componentes funcional, segurança e confiabilidade, desempenho, operação e manutenção. Estes requisitos são desagregados em requisitos de baixo nível para os quais é proposta uma arquitetura de sistema com módulos funcionais e não funcionais. No caso específico dos requisitos de segurança e confiabilidade, foi realizado um levantamento das ameaças e vulnerabilidades à componente aplicacional do sistema, com o objetivo de identificar os controlos necessários e propor um conjunto de componentes que, sendo eles próprios conformes, garantem conformidade com os controlos identificados. A análise inicia-se com a identificação dos ativos relevantes, a que se segue a identificação das ameaças e vulnerabilidades correspondentes, com maior foco nas ameaças para a aplicação e na ameaça que esta, se e

quando comprometida, pode constituir para a rede inteligente. Dos controlos identificados, são apenas incluídos no desenho aqueles que têm de ser implementados através de componentes aplicativos ou para os quais a aplicação tem de dar algum tipo de suporte. Os controlos externos não são cobertos por esta investigação.

Ainda sobre o desenho funcional, é feito um modelo da rede inteligente, incluindo os sistemas e componentes das suas várias camadas, com o objetivo de identificar as configurações que cada um suporta e as ligações entre eles. São também modelados, com o objetivo de identificar ligações e dependências: o processo de operação da rede elétrica, um processo genérico representativo dos processos e serviços dependentes do estado operacional da rede elétrica e o processo de tratamento automático. Estes modelos são utilizados na fase de implementação.

A arquitetura resultante é a de um sistema multi-agente com agentes geograficamente distribuídos e replicados, designados por entidades especialistas em tratamento de faltas. Cada entidade é responsável por um domínio de tratamento limitado, correspondendo a um conjunto de sistemas, componentes e serviços da rede inteligente que fazem parte do seu âmbito de supervisão. Raciocina sobre conhecimento assente em factos e regras. Supervisiona o seu domínio, diagnosticando faltas, criando planos de recuperação e reconfigurando a rede inteligente com base nesses planos. Cooperar com outras entidades. Aprende com os resultados e consequências da sua atuação. Integra componentes de segurança e confiabilidade para prevenir e tolerar faltas e intrusões nos seus próprios componentes.

O sistema é implementado parcialmente para prova do conceito. A implementação inclui a definição de um domínio de tratamento, da ontologia correspondente, do modelo de conhecimento com factos e regras, dos objetivos de tratamento e de um conjunto de *queries* aplicáveis ao modelo. O domínio de tratamento inclui componentes da rede elétrica, equipamentos de rede, computadores e um sistema de controlo de acessos físicos, cobrindo desta forma diferentes camadas da rede inteligente. Para validação da implementação, os objetivos e *queries* são submetidos a um motor de inferência, no qual o modelo de conhecimento é previamente carregado, simulando o comportamento de uma réplica nos diferentes estados do processo de tratamento. O processo é repetido para quatro cenários de faltas e falhas de complexidade crescente, incluindo um cenário de falta de conhecimento em que o resultado da inferência, demonstrando a necessidade de manter as bases de conhecimento atualizadas.

A implementação dos restantes módulos e integração do módulo de conhecimento é deixada para trabalho futuro, o que limita a validação da segurança da solução. Por definição, os controlos incluídos na arquitetura proposta respondem aos requisitos do sistema, dado que o desenho da solução utiliza módulos de segurança e confiabilidade identificados através de uma análise de ameaças e vulnerabilidades. No entanto, a verificação de que estes controlos são corretamente implementados e a validação da robustez dessa

implementação está dependente da implementação dos módulos e, por esta razão, é deixada também para trabalho futuro.

Validamos também a robustez do desenho proposto em termos de *liveness* e *safety*. Neste sentido, apresentamos uma definição para cada uma destas propriedades no contexto da solução proposta, apresentamos um conjunto de cenários em que as mesmas são comprometidas e justificamos o porquê de esses cenários não serem possíveis. No caso da *liveness*, o sistema deve executar continuamente desde a sua instalação até ao fim do seu ciclo de vida, entre eventuais interrupções programadas. Para a sua validação focamo-nos nas interações entre os vários módulos, com os sistemas e componentes da rede inteligente e entre entidades. No caso da *safety*, as ações do sistema devem basear-se apenas em informação atualizada, recolhida dos sistemas e componentes da rede inteligente. Neste caso, o foco é no conteúdo do modelo de conhecimento, na coordenação entre réplicas e a execução de comandos nos sistemas e componentes da rede inteligente.

Por último, discutimos um conjunto de tópicos de desenho e implementação que, sendo críticos para a segurança e robustez do sistema proposto, dependem do contexto específico da cada rede inteligente e fornecemos recomendações e orientações para os mesmos. Assumindo a existência de outros sistemas instalados na rede inteligente com atuação possivelmente concorrente com a aqui considerada, é necessário definir qual é o âmbito de cada um esse haverá ou não interação entre o sistema aqui proposto e esses sistemas. O sistema aqui proposto poderá utilizar os sensores, atuadores e redes de comunicações já existentes, dependendo de garantias funcionais, desempenho, capacidade e segurança dados pelos mesmos, para adquirir a informação necessária e controlar os sistemas e componentes da rede inteligente, sendo necessário identificar as necessidades de implementação associadas. A alternativa é construir completa ou parcialmente uma infraestrutura dedicada. Este sistema poderá ser criado de raiz ou a partir de outros sistemas já existentes e que contenham módulos com funcionalidades semelhantes às identificadas no desenho da solução. É necessário instalar, operar e manter o sistema com o conhecimento necessário à tomada de decisão. Se tal não for feito, o sistema poderá tomar decisões prejudiciais. A distribuição do sistema, em termos de número de domínios, e a sua replicação, em termos de número de réplicas, tendo previsivelmente um impacto elevado nos custos da solução, deverão ter em conta análises de risco e de custo-benefício. Uma distribuição com granularidade apropriada e um número suficiente de réplicas com distribuição adequada permitem que o sistema funcione corretamente também em casos de partição de comunicações e/ou conectividade. As decisões tomadas, relacionadas com estes tópicos, têm impacto direto no desempenho, segurança e confiabilidade da solução.

Para trabalho futuro, a nível de desenho, é proposto: a evolução de alguns módulos já incluídos no desenho da solução e o desenvolvimento de novos módulos, a modelação de mais sistemas, componentes e serviços e a atualização e extensão da análise de ameaças e vulnerabilidades. A nível de implementação, é proposto: a formalização e manutenção de

uma ontologia de suporte à descrição dos sistemas, componentes e serviços, a atualização dos factos, com base na ontologia, e a melhoria das regras, aproximando-as incrementalmente da realidade, o desenvolvimento do código de software associado a cada módulo e a extensão das recomendações e orientações apresentadas na discussão para incluírem exemplos práticos.

Palavras-chave: redes inteligentes, tratamento automático de faltas, sistemas inteligentes, segurança, confiabilidade

Abstract

The increasing complexity of the smart grid raises concerns with performance, privacy, security and dependability that go further beyond electrical network faults. In this regard, electrical network self-healing and commercially available security solutions are capable of handling a set of electrical network, systems and communications faults automatically, but separately. However, as shown by the Ukrainian incidents, in 2015, there can be cause-effect connections between faults and failures in different smart grid layers. Additionally, although a set of European projects is addressing the security and dependability of self-healing use cases, the pilot projects focus mainly on functional issues, possibly compromising the security of future roll-outs.

We use a knowledge-based and security-by-design approach to design and propose a secure and dependable Self-Healing System (SHS) with awareness of the aforementioned connections. It is a Multi Agent System (MAS) with replicated Self-Healing Expert Entity (SHEE) agents. Each SHEE is responsible for the self-healing process in a limited domain, corresponding to a set of systems, components and processes assigned to its scope of supervision. It reasons with knowledge based on facts and rules. It monitors the domain, diagnoses eventual faults, creates recovery plans and reconfigures the smart grid based on these plans. It cooperates with other SHEEs. It learns from the results and consequences of its actions. It comprises a set of security and dependability features to prevent and tolerate faults and intrusions, resulting from a threat and vulnerability assessment.

We perform a partial implementation of our system, consisting in the definition of a self-healing domain, the corresponding ontology, the knowledge model with facts and reasoning rules and a set of goals and queries. We successfully validate the SHS concept as a solution to the described problems. The goals and queries are submitted to a standalone inference engine, which is previously loaded with the knowledge model, simulating the behavior of a SHEE replica through the different states of the self-healing process. The process is repeated for four different complexity increasing fault and failure scenarios. We discuss and provide guidance for a set of design and implementation issues that, being critical to the security and robustness of the SHS, depend on each smart grid specific context.

Keywords: smart grids, self-healing, expert systems, security, dependability

Contents

List of Figures	xvi
------------------------	------------

List of Tables	xvii
-----------------------	-------------

1 Introduction	1
1.1 Problem	1
1.2 Hypothesis	1
1.3 Scope of work	2
1.4 Objectives and Contributions	2
1.5 Work Overview	2
1.5.1 Context	2
1.5.2 Design	3
1.5.3 Implementation	5
1.6 Document Structure	7
2 Context	9
2.1 Chapter Overview	9
2.2 The Electricity System Value Chain	10
2.2.1 Generation	11
2.2.2 Transmission	12
2.2.3 Distribution	12
2.2.4 Supply	13
2.3 Classic and New Challenges for DSOs	13
2.3.1 Energy Efficiency	14
2.3.2 Renewable Energy Sources	14
2.3.3 Electric Vehicles	15
2.3.4 New Business Models	15
2.4 Smart Distribution Grids	16
2.4.1 Facilities and Cabinets	16
2.4.2 Electrical Network	17
2.4.3 Sensors and Actuators	18

2.4.4	SCADA and DMS	18
2.4.5	Substation Automation	19
2.4.6	Distribution Automation	20
2.4.7	Advanced Metering Infrastructure	20
2.4.8	External Connections	21
2.4.9	Communications	21
2.4.10	Microgrids	22
2.5	Smart Grid Concerns	22
2.5.1	Performance	23
2.5.2	Privacy	23
2.5.3	Security	23
2.5.4	Dependability	24
2.6	Automation and Self-healing Approaches	25
2.6.1	The EDP Distribuição Case	25
2.6.2	The Stedin Case	28
2.6.3	The SEGRID Project Case	28
2.6.4	The e-balance Project Case	29
2.7	Cyber Security and Dependability Controls	31
2.7.1	Currently Used Solutions	32
2.7.2	Relevant Research and Development	35
3	Design	39
3.1	Chapter Overview	39
3.2	Problem Definition	41
3.2.1	Requirements	42
3.2.2	Assumptions and Constraints	42
3.2.3	Hypothesis Refinement	43
3.3	Functional Design	43
3.3.1	Requirements Breakdown	43
3.3.2	Intelligent Supervision System Components	44
3.3.3	Expert System Components	45
3.3.4	Knowledge Modeling	46
3.3.5	Information Collection and Control	47
3.3.6	User Interaction	47
3.4	Smart Grid Model	47
3.4.1	Electrical Network	48
3.4.2	Communications Network	48
3.4.3	SCADA and Automation	51
3.4.4	Further Considerations	53
3.5	Processes Model	54

3.5.1	Smart Grid	54
3.5.2	Smart Grid Dependent Processes	56
3.6	Security and Dependability Design	57
3.6.1	Threat and Vulnerability Assessment	57
3.6.2	Controls	59
3.6.3	Distributed Application	61
3.6.4	Self-healing Domains	63
3.6.5	Security Features	64
3.6.6	Dependability Techniques	64
3.7	A Self-healing Expert Entity	65
3.7.1	Architecture	65
3.7.2	Functional Description	70
3.7.3	Non-functional Description	76
4	Implementation	79
4.1	Chapter Overview	79
4.2	Proof of Concept	81
4.2.1	Self-healing Domain	81
4.2.2	Knowledge Modeling	83
4.2.3	Goals and Queries	87
4.3	Functional Validation	88
4.3.1	Methodology	88
4.3.2	Scenario 1: A Firewall Reconfiguration	90
4.3.3	Scenario 2: The Consequences of an Incomplete Knowledge Base	91
4.3.4	Scenario 3: Physical Security and SCADA	92
4.3.5	Scenario 4: An Electrical Failure in a Compromised Infrastructure	94
4.4	Security Validation	97
4.5	Robustness Validation	97
4.5.1	Liveness Validation	97
4.5.2	Safety Validation	99
4.6	Discussion	101
4.6.1	Creating a Self-healing Ecosystem	102
4.6.2	Using the Existing Infrastructure in the SHS	102
4.6.3	Adapting Existing Systems to Create a SHEE	104
4.6.4	Restrictions and Limitations to the Self-healing Process	105
4.6.5	Distributing the SHS and Replicating the SHEEs	105

4.6.6	Closing Remarks	108
5	Conclusion	111
5.1	Results	111
5.2	Future Work	112
5.2.1	Design	113
5.2.2	Implementation	113
	Acronyms	119
	Bibliography	126
A	Table of Threats, Vulnerabilities and Controls	127
B	POC Knowledge	133
B.1	Facts	133
B.2	Rules	133

List of Figures

1.1	Self-healing System.	5
2.1	Electric System.	11
3.1	Information flow in a supervision architecture.	45
3.2	Electrical network model.	49
3.3	Communications network model.	50
3.4	SCADA and automation.	52
3.5	Smart grid operation model.	55
3.6	Generic smart grid dependent process model.	56
3.7	Self-healing multi-agent system.	67
3.8	SHEE Architecture.	68
3.9	Self-healing process model.	69
3.10	SHEE knowledge and reasoning.	71
4.1	Self-healing domain.	82
4.2	Detailed self-healing domain components.	84
4.3	Self-healing cycle.	89
4.4	Scenario 1 results.	91
4.5	Scenario 2 results.	92
4.6	Scenario 3 results.	94
4.7	Scenario 4 results.	96
4.8	Self-healing system without network partitions.	107
4.9	Network partition examples.	109
B.1	Router 1 facts.	134
B.2	Substation controller 1 facts.	135
B.3	HMI 1 facts.	136
B.4	Switch 2 facts.	136
B.5	Firewall 1 facts.	137
B.6	Switch 1 facts.	137
B.7	IED 9 facts.	138
B.8	Electrical switch 9 facts.	138

B.9 Door 1 facts.	139
B.10 SHEE2 facts as seen from SHEE1.	139
B.11 Switching rules.	140
B.12 Routing rules.	140
B.13 Firewall rules.	141
B.14 Authentication rules.	142
B.15 Substation controller operating mode rules.	143
B.16 IED operating mode rules.	144
B.17 Electrical switch rules.	145
B.18 Rule to retrieve the configurations associated with a component interface.	145
B.19 Rule to cycle through the service dependencies associated with a connection.	146
B.20 Rule to cycle through the connections associated with a service.	146
B.21 Rule to retrieve the configurations associated with a service.	147
B.22 Rules associated with the monitoring activity.	147
B.23 Rules associated with the diagnosis activity.	148
B.24 Rules associated with the recovery activity.	149
B.25 Rule to concatenate a list of lists.	149
B.26 Prolog initializations.	149

List of Tables

3.1	Assets table.	58
4.1	Self-healing communications infrastructure deployment comparison.	103
4.2	Distributed deployment comparison.	106
4.3	Replication deployment comparison.	106
A.1	Threat and vulnerability assessment.	127

Chapter 1

Introduction

Stories are told in books and films about how, thousands of years ago, we learned to control fire, a process through which the burning matter releases energy that we can use for cooking food, heating our homes, protection against predators and lighting the way. Since then, we have learned how to generate electrical energy from many different energy sources such as coal, natural gas, water, wind or solar radiation. We found ways to transport and distribute this energy across countries, down from where it is generated, and supply it to where it is consumed. The electrical network grew, reaching further and farther into our homes, work and public places and creating an increasing dependence as increasingly more activities and services deeply rely on it. Currently, this network is a critical infrastructure that, in case of incapacitation or destruction, would have a debilitating effect on the economy and public safety. It is supervised by operators with support from complex systems and communication networks, which, together with the electrical network, comprise a smart grid.

1.1 Problem

These smart grid systems are becoming increasingly intelligent and autonomous. Nevertheless, the information they generate, the decisions they make and the actions they take are restricted by the information they have, which can make them deceive the operators and take harmful actions. Additionally, they are still vulnerable to security threats, which might make them behave erroneously or maliciously, with possible harmful consequences to the smart grid and everything and everyone who relies on it.

1.2 Hypothesis

It is possible to make a knowledge-based, secure and dependable system that, by leveraging from various kinds of information for decision making, can contribute to improve the overall healing abilities of the smart grid.

1.3 Scope of work

We address the problem and research hypothesis for the case of smart grid electricity distribution grids and, more specifically, the Self-Healing Systems (SHSs) solution. Smart grid self-healing, security and dependability, together or separately, are the focus of several current European projects and initiatives, being main concerns for the future smart grid. Self-healing was chosen, among other smart grid advanced functionalities, due to the requirement of horizontal communications between devices and automatic reconfigurations, raising a distinct set of functional, security and dependability concerns. Additionally, as new and advanced self-healing solutions are still being researched and developed, we are still within the optimum time frame to propose improvements regarding the aforementioned concerns.

1.4 Objectives and Contributions

The objective and contribution of this thesis is to design and prove the concept of a knowledge-based, secure and dependable SHS, in compliance with the hypothesis in Section 1.2. The results can be used to implement the SHS and/or to improve the functionalities, security and/or dependability of current and new smart grid intelligent and autonomous systems.

1.5 Work Overview

1.5.1 Context

The electricity distribution network is an electrical infrastructure that takes the electricity produced in centralized generation (e.g., dams and gas power plants), distributed generation (e.g., wind farms) and Micro-Generation (MG) (e.g., rooftop solar panels) and brings it to our homes and to many other society infrastructures that we depend on every day. Its incapacitation or destruction would have a debilitating effect on the economy and public safety, which has earned it the critical infrastructure classification.

Through the years, the increasing needs for electrification and power made it grow to a country-wide size. In its evolution, the integration of new types of generation and loads and the increasingly strict efficiency and Quality of Service (QoS) regulations made the electrical infrastructure change from a locally and manually controlled electrical network, which was completely isolated, to an automated and remotely controlled, highly connected smart grid. Local and central control systems and a country-wide communications network were layered on top and connected to the electrical network to support core functionalities, such as Supervisory Control and Data Acquisition (SCADA) and substation automation, and advanced functionalities, such as Distribution Automation (DA) and

Advanced Metering Infrastructure (AMI).

Performance, privacy, security and dependability are key smart grid concerns, which are related with the handling of real-time data and sensitive information. In addition, it is a proven fact that the smart grid is exposed and vulnerable to different threats, ranging from the "traditional" electrical faults to malicious attacks in the systems and communications. Substation automation and DA have the capability to handle faults occurring in the electrical network, automating incident response to distinct lengths of the Fault Location, Isolation and Restoration (FLIR) sequence for different voltage sections. This self-healing behavior is the current scope of a set of world-wide deployments and Research and Development (R&D) initiatives, mainly focusing in its functional aspects. Security related faults in systems and communications are handled with the support of a set of commercially available solutions, such as firewalls, Intrusion Detection Systems (IDSs), antivirus and Security Information and Event Management (SIEM) systems, which need to comply with the smart grid performance requirements. These solutions are usually based on a predefined group of rules, signatures or use cases, which automate intrusion prevention and detection for a set of specific cases.

1.5.2 Design

The electrical network self-healing and security automation solutions are restricted in the decisions they make and in the actions they take, since they operate based on information about a specific smart grid technical layer, disregarding related information from the other layers. In this respect, we propose a SHS that acquires and reasons with knowledge from the several smart grid layers to create more efficient, accurate and robust failure prevention and recovery plans, while minimizing their impact in the smart grid performance. The plans can be automatically executed, by sending commands to the corresponding smart grid systems and components, or they can be proposed to the smart grid operators. Intelligent supervision system components are used to provide monitoring, diagnosis, recovery and learning capabilities. Expert system components provide knowledge management and reasoning abilities. Ontologies provide knowledge modeling capabilities. We model the systems and components in the several smart grid layers, focusing on their connections, functionalities and emphasizing the relationships between the layers. We model the smart grid and its dependent processes as state machines that change state with fault and failure events and also with their reconfiguration in response to these events.

The self-healing pilot projects are mainly focused on functional issues, which might make them disregard security-by-design benefits, compromising the overall security of eventual future roll-outs. In this regard, we employ a security-by-design approach to identify and propose the security and dependability requirements that provide the required protection, intrusion and fault tolerance capabilities to the SHS, which is motivated by the access to confidential information and the capability of automatically control the

smart grid. We perform a threat and vulnerability assessment to the SHS, focusing on the self-healing application as a target and as a possible threat to the smart grid, behaving erroneously or maliciously. The corresponding controls are identified and a subset comprised of those that need to be technically addressed by the SHS is proposed in the form of requirements to the security and dependability components.

A Multi Agent System (MAS) architecture with hybrid vertical "two-pass control" layered and cooperating agents provides distributed self-healing capabilities. In partitioning scenarios, the agents are capable of supervising the smart grid systems and components that remain within the same partition. The assignment of each agent to a self-healing domain provides role segregation within the MAS. A set of security features, including cryptography, user management, event logging, backup and configuration modules, provide secure access, input/output (I/O), communications, storage, event logging, backup and configuration capabilities. Byzantine Fault-Tolerant (BFT) State Machine Replication (SMR), proactive and reactive recovery of replicas provide tolerance against faults and intrusions. One of the possible protocols is MinBFT [1], requiring at least $2f + 1$ replicas to tolerate f faults.

The proposed functional, security and dependability components are materialized in the Self-Healing Expert Entity (SHEE) architecture and corresponding modules. A SHEE is a replicated, intelligent, autonomous and cooperating agent of the multi-agent SHS. It is responsible for an assigned self-healing domain, which consists of a set of the smart grid systems, components and processes that are in its scope of supervision. Figure 1.1 depicts a set of SHEEs with different self-healing domains and the expected connections between one-another and with the supervised systems and components. There is a replicated SHEE in each primary and secondary substation. The SHEEs at primary substations 1 and 2 supervise not only the primary substation but also the Medium Voltage (MV) electrical network and corresponding DA devices. The green lines represent the expected connections between the SHEEs having service related self-healing domains and with the supervised systems and components.

A SHEE reasons with knowledge based on facts and rules. It monitors the self-healing domain, detecting eventual faults and failures. It diagnoses eventual faults to determine the cause and impact of occurred failures and to find possible causes for future failures. It creates recovery plans, consisting in an ordered sequence of fault removal and/or isolation actions, to recover the smart grid from the observed failures and to prevent future failures. It reconfigures the smart grid based on the aforementioned plans. It learns from the past decisions and from the results and consequences of its actions. It communicates with other SHEEs in the MAS by using standard-based protocols and a standard-based language. It comprises a set of security and dependability features to prevent and tolerate faults and intrusions. It has a hybrid vertical "two-pass control" layered architecture, comprising several modules, such as the knowledge, monitoring, diagnosis and recovery modules.

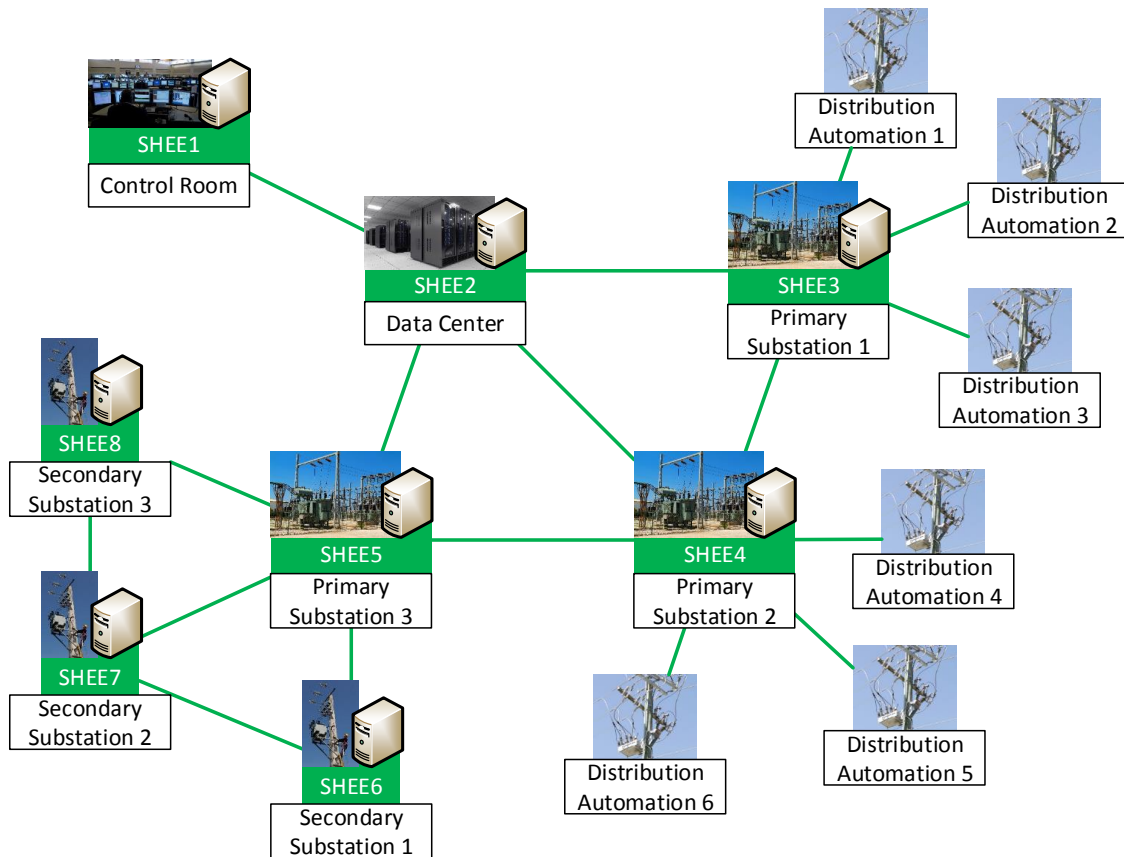


Figure 1.1: Self-healing System.

1.5.3 Implementation

A SHEE is defined by the assigned self-healing domain, the monitoring capabilities, the knowledge, the goals and queries and the control capabilities. In this regard, the Proof of Concept (POC) implementation includes the following steps:

1. The selection of the self-healing domain, which comprises a set of primary substation components from the electrical, the control, the communications, the physical security and the cyber security layers, which support the electricity distribution process and the local and remote operation of the substation. This set was chosen for its diversity in terms of included smart grid processes and architecture layers, which lets us focus on a well-delimited test case, while allowing to draw conclusions that expectedly apply also to other domains that contain the same set or a subset of similar processes and layers.
2. The components are mapped to classes, properties and individuals of an ontology, from which we extract knowledge facts to create the knowledge model. The facts reflect the existing relations between the ontology individuals (i.e., the domain components) from the moment when they are asserted to the knowledge model until the moment when they are removed.

3. The SHEE attempts to solve a connectivity problem, attending to the cyber-physical connectivity that enables the availability of the electricity distribution process and of the local and remote operation of the substation. For this purpose, we define and assert reasoning rules that enables iterations over the domain components to verify the existence of connectivity enabling operational conditions, while modeling their behavior.
4. We define the goals and queries which are used by the monitoring, diagnosis, recovery, reconfiguration, cooperation and learning agent behaviors to query the knowledge model.

We successfully validate the SHS concept. For this purpose, we make three slightly modified versions of the developed knowledge model, corresponding to different fault and failure scenarios. The models are loaded into a standalone Prolog inference engine - SWI Prolog - to which the goals and queries are submitted in an ordered sequence, simulating the behavior of a SHEE replica through the different states of the self-healing process, for each scenario. In this regard, only the knowledge model code was implemented in the scope of the POC. The software code for the knowledge module and remaining modules is proposed for future work. The robustness of the proposed solution is validated by explaining how it prevents a set of liveness and safety compromising scenarios.

Still within the scope of the current work, there is a set of design and implementation issues that, being critical to the security and robustness of the SHS, depend on each smart grid specific context. In this regard, we provide guidance for the following:

- The SHS must be separated or integrated with existing electrical network self-healing and security automation solutions, avoiding concurrency situations and fostering synergies that are only possible through their integration.
- It can take advantage of the existing smart grid infrastructure, namely, of the existing communications infrastructure, information repositories and smart grid controllers, avoiding the deployment of a self-healing specific communications infrastructure and the direct connection with the sensors and actuators, which might require the implementation of certain safety measures.
- It might be possible to implement a SHEE starting from the existing electrical network self-healing and security automation solutions, which demonstrate some of the supervision behaviors required by the SHEE modules and are already capable of collecting the required information and controlling certain smart grid components.
- It is limited in its decisions and actions by the available resources, which include the monitoring and control capabilities, the knowledge and the goals and queries.

Therefore, the design and implementation of the required collectors and controllers, the use of secure and dependable communication channels and the definition of an enabling knowledge representation, populated with the required facts, rules and queries, are essential to a successful execution of the self-healing process.

- Its correct execution depends on its distribution granularity, on an adequate number of replicas per SHEE and on a proper replica distribution. Therefore, these steps should ensure an acceptable risk to the smart grid network locations, regarding partitioning, and to the unique points of failure between SHEE replicas.

The proper design and implementation decisions contribute to a successful and correct execution of the self-healing process by the SHS. They also provide relevant information regarding the cause and impact of failures that might be used to reduce incident response times and to prevent incident recurrences.

1.6 Document Structure

This document is organized as follows:

- Chapter 2 - Describes the supporting concepts of the work, with the purpose of providing context for the solutions that are presented, analyzed and discussed in following chapters;
- Chapter 3 - Proposes a knowledge-based, secure and dependable SHS architecture for smart grids;
- Chapter 4 - Describes a POC implementation, including its validation and a discussion of the overall solution;
- Chapter 5 - Presents the final results and lists a set of remaining issues to be handled in future work.

Chapter 2

Context

This chapter describes the supporting concepts of the work, with the purpose of providing context for the solutions that are presented, analyzed and discussed in following chapters. This chapter is organized in the following way:

- The Electric System Value Chain - Presents the main roles in the electric system value chain;
- Classic and New Challenges for DSOs - Explains how the electric system value chain is changing, focusing on the challenges to the Distribution System Operator (DSO);
- Smart Distribution Grids - Presents the smart distribution grid as an answer to the changing context, including its core functionalities and components;
- Smart Grid Concerns - Discusses the performance, privacy, security and dependability concerns raised by the integration of computer systems and communications with the electrical network;
- Automation and Self-Healing Approaches - Presents the different types of automation and self-healing approaches that are currently used to handle electrical faults;
- Cyber Security and Dependability Controls - Presents the cyber security and dependability controls that are currently used in smart grids and relevant new solutions that are being proposed by R&D works.

2.1 Chapter Overview

The electricity distribution network is an electrical infrastructure that takes the electricity produced in centralized generation (e.g., dams and gas power plants), distributed generation (e.g., wind farms) and MG (e.g., rooftop solar panels) and brings it to our homes,

schools, hospitals, factories, ports, airports, communications, finance, water and wastewater systems and to many other society infrastructures which depend on it every day. Its incapacitation or destruction would have a debilitating effect on the economy and public safety, which has earned it the critical infrastructure classification. Through the years, the increasing needs for electrification and power made it grow to a country-wide size, while the integration of new types of generation and loads and the increasingly strict efficiency and QoS regulations made it change from a locally and manually controlled electrical network, which was completely isolated from any kind of computers or communications network, to an automated and remotely controlled, highly connected smart grid. Local and central control systems and a country-wide communications network were layered on top and connected to the electrical network to support core functionalities, such as SCADA and substation automation, and advanced functionalities, such as DA and AMI. Performance, privacy, security and dependability are key smart grid concerns, which are related with the handling of real-time information and sensitive information and with the proven fact that it is exposed and vulnerable to different threats, ranging from the "traditional" electrical faults to non-malicious faults in the systems and communications and to the more recent cyber attacks. Substation automation and DA have the capability to handle faults occurring in the electrical network, automating incident response to different lengths of the FLIR sequence for different voltage sections. This self-healing behavior is the current scope of a set of world-wide real-world deployments and R&D initiatives, mainly focusing in its functional aspects. The faults in systems and communications are handled with the support of a set of commercially available security solutions, such as firewalls, IDS, antivirus and SIEM, which need to comply with the smart grid performance requirements. These solutions are usually based on a predefined set of rules, signatures or use cases, which automate intrusion detection and prevention for a set of specific cases. A set of European Projects and published works are researching and developing security and dependability solutions for the future smart grid systems and communications, also considering a set of electrical self-healing use cases.

2.2 The Electricity System Value Chain

In Europe, the electricity system value chain is comprised by the activities: generation, transmission, distribution and supply of electricity, as depicted in Figure 2.1 [2]. A contextual description for each function is provided beneath, with its focus being placed on how each function relates with electricity distribution, which is the the main scope of this dissertation (see Section 1.3).

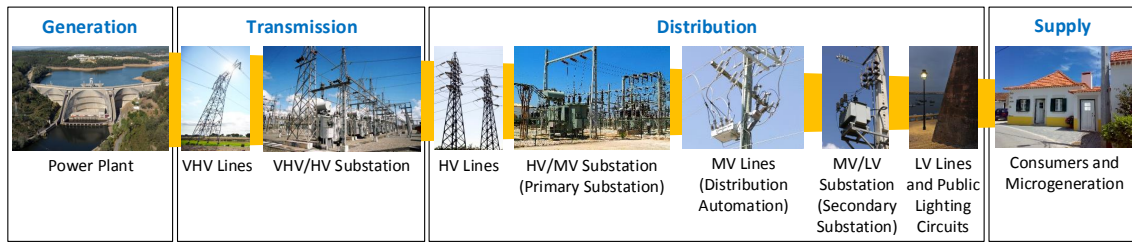


Figure 2.1: Electric System.

2.2.1 Generation

Generation is the activity in which electricity is generated with the use of a diverse set of technologies from different primary energy sources.

Renewable and Non-renewable Energy Sources

Energy sources that cannot be replenished sufficiently fast to allow a sustainable exploration within acceptable human time-frames are called non-renewable energy sources; e.g., fossil fuels, such as coal and natural gas, and uranium fit in this category (BP has published a statistical review of existing reserves and production of non-renewable energy sources, including reserves versus production ratios that indicate how long these reserves will last [3]). On the other hand, energy sources that are constantly being replenished are called Renewable Energy Sources (RES); e.g., water, wind, solar radiation, biomass and biogas fit in this category. If we are able to respect and take care of planet Earth, RES will continue to exist until long after fossil fuels' reserves are depleted.

Renewable and non-renewable energy sources have different impacts in electricity distribution. The exploration of non-renewable sources allows the creation of reserves, which are constantly being used and refilled. If the reserves are adequate to a specific generation process and associated risks, there should be uninterrupted availability, which translates to a constant electricity flow from the power plant to the consumers, according to plan. RES are not always available; e.g., there might not be enough water after a drought year, wind might stop blowing during the day and there is no sun during the night or, even during the day, as clouds can block most of the solar radiation. Therefore, RES related electricity flows are more difficult to forecast and manage.

Centralized Generation and Distributed Generation

Centralized generation is used when the energy source exists or is made available only at a specific set of geographic locations; e.g., it is the case of non-renewable energy sources, which requires the transport and distribution of electricity between the geographic locations of generation and supply, as far away from each other as it takes to go from one country to the another in some situations. On the other hand, when the energy source is

geographically distributed, such as RES, and depending on the availability of adequate technologies, Distributed Generation (DG) can be used either for local consumption or to supply electricity to the power grid at transport or distribution levels.

Centralized generation and DG also have different impacts in electricity distribution. In the case of centralized generation, the construction of a new power plant typically includes a long-framed structuring project; e.g., as it happens with thermal coal and gas, and big hydroelectric power plants, which occur less frequently than DG. Wind farms and MG with photo-voltaic panels are common examples of DG. While wind farms can be seen as a middle step between a big power plant and MG, given the afore mentioned context, it is the last that causes the most impact, as it is explained below.

Consumers are able to turn into prosumers by installing MG solutions at their homes. The generated electricity can be consumed at the home where it is produced or, if it is not needed there, it can be supplied to the electrical network. In the latter case, if a lot of electricity is generated at locations where it is not needed, it has to be distributed to the locations where it is needed, which can be, for instance, next door or in the next city. If the electrical network is not prepared, such events may induce excess currents in the distribution lines, which can cause the lines to overheat and degrade more rapidly. Their life expectancy is, therefore, reduced.

2.2.2 Transmission

The electricity generated in power plants is transported to the supply locations through the transmission network, which is performed in Very High Voltage (VHV) to minimize or reduce energy losses. The actors that are responsible for this activity are the Transmission System Operators (TSOs).

It is common for TSOs and DSOs to exchange information related with power grid state about the areas where the transmission delivers electricity to the distribution grid. This information is used by both sides as an extra valuable input for planning operation and management of the power grid.

In cases where this information is exchanged through communication links between the systems of both actors, the link must be seen as a potential source of failures and, therefore, it should be included in the risk analysis of the corresponding organizations.

2.2.3 Distribution

The electricity delivered by the transmission grid is distributed through the electricity distribution grid - primary substations, secondary substations and lines - to consumers. In this process, electricity is transformed to lower voltage levels in substations to High Voltage (HV), MV and Low Voltage (LV). The distribution grid also distributes the electricity generated at DG power plants and MG installations, which are integrated in the

distribution grid. The actors that are responsible for this activity are the DSOs.

2.2.4 Supply

Electricity suppliers are responsible for billing the electricity to end customers, to whom they may also offer energy services. To perform these activities, they require electricity consumption data from customers, which can be acquired through the traditional way or through smart metering. The traditional way consists of periodically sending people to each consumption location to collect the corresponding meter readings. Smart metering allows the remote collection of meter readings on a given schedule together with a whole set of new functionalities and use cases, which are further explained in Section 2.4.7.

The handling of smart metering data, together with other smart grid data, is currently the focus of an ongoing work at European Commission (EC) level. This work, which is further explained in Section 2.3.4, circles around which entities should collect, process, communicate and store the data and how they should interact with each other. An option would be for the DSO to collect this data and to supply it to the electricity suppliers. In Portugal, for example, it is in the responsibilities of the DSO to deploy and maintain the smart metering infrastructure and to collect the smart metering data.

2.3 Classic and New Challenges for DSOs

During a time when Europe was seeing a continuous growth in electricity consumption, the main challenge for DSOs was the expansion of the electricity distribution power grid in a sustainable manner. SCADA systems were then used by DSOs to enable monitoring and control of the rapidly growing number of power equipment as a means to supply customers with high QoS and achieving operational efficiency.

Times have changed with the coming of a new financial crisis that led to less electricity consumption, with negative impact in already existing businesses, some of which stopped growing, became smaller or closed doors, and the survival of new businesses. For all the electricity system value chain actors and activities, this translates to less energy being supplied, distributed, transported and generated (in the specific case of electricity supply, see [4], which shows the electricity supply evolution in European Union (EU) Member States from 2000 to 2014). In the case of DSOs, the effort is being directed to optimizing the distribution grid, with new substations and lines are still being constructed as a means to reinforce or provide redundancy to needing geographical location. In this new context, rising the QoS and maximizing operational efficiency are still up-to-date challenges. Nevertheless, DSO are facing new challenges in the areas of energy efficiency and new business models, integration of RES, DG and Electric Vehicles (EVs).

2.3.1 Energy Efficiency

Energy efficiency is a EC policy area within the EU's Energy Union Strategy for its Energy Union and Climate priority [5]. Its goal is to consume "less energy in order to reduce pollution and preserve domestic energy sources", which will "reduce the EU's need for energy imports" [6]. To promote energy efficiency, the EC has included it in its targets for 2020 and 2030. According to the 2020 Energy Strategy, energy efficiency should be improved by at least 20%. Also, EU countries have agreed to increase energy efficiency by at least 27% by 2030 [7, 8].

All the activities of the electricity system value chain have a role to play on the road to energy efficiency, from the decarbonization of the Generation, through the reduction of losses in the Transmission and Distribution and to the reduction of demand and its redistribution over time. Regarding the last, Demand Side Management (DSM) programs are used to attain long-term reductions in demand by encouraging the consumers to use electricity in a more efficient manner, while Demand Response (DR) programs are used to encourage the consumers to make short-term reductions in electricity demand in response to a price signal from the electricity hourly market or a trigger initiated by the electricity grid operator. Smart metering infrastructures, which are further detailed in Section 2.4.7, play a role in this programs by giving the necessary technology support and enabling electricity consumers to have a better understanding of their consumption habits and being able to change them.

2.3.2 Renewable Energy Sources

The development of RES is also included in EC's key policy areas for its Energy Strategy. The 2020 Energy Strategy set as target to "increase the share of renewable energy in the EU's energy mix to at least 20% of consumption". Also, EU countries have agreed to "a binding target of at least 27% of renewable energy in the EU" [7, 8]. In the Energy Roadmap 2050 it is stated that "renewables will move to the center of the energy mix in Europe, from technology development to mass production and deployment, from small-scale to larger-scale, integrating local and more remote sources". In the same report, it is also stated that "storage technologies remain critical" to enable the switching to RES [9]. Within European legislation, Directive 2009/28/EC states that "Member States shall take the appropriate steps to develop transmission and distribution grid infrastructure, intelligent networks, storage facilities and the electricity system, in order to allow the secure operation of the electricity system as it accommodates the further development of electricity production from renewable energy sources, including interconnection between Member States and between Member States and third countries" [10].

RES-related DG and MG brings challenges to electricity distribution operation and management due to the inconsistency of generated electricity flows, as it was explained

in Section 2.2.1.

2.3.3 Electric Vehicles

The Energy Roadmap 2050 states that "another area of special importance is the shift towards alternative fuels, including electric vehicles < . . .>batteries, fuel cells and hydrogen, which together with smart grids can multiply the benefits of electro-mobility both for decarbonization of transport and development of renewable energy" [9]. Directive 2014/94/EU states that "electricity has the potential to increase the energy efficiency of road vehicles and contribute to a CO₂ reduction in transport". The Directive also charges Member States for ensuring that a publicly accessible EVs supply infrastructure is put in place, including a number of recharging points that is adequate to the number of expected registered EVs by 2020 and that can be normal or high power. Recharging can make use of intelligent metering systems to enable functionalities such as off-peak charging or vehicle-to-grid electricity flows during high electricity demand periods. These functionalities optimize recharging, benefit the customer and ensure the stability of the grid [11].

Today, the ratio of autonomy and charging times over vehicle and battery costs is still too low to allow a general purpose large-scale adoption of electric vehicles. Nevertheless, Joint Research Centre (JRC) reports in 2015 say that the number of electrical vehicles' sales is increasing by year. Statistics show that the number of sold cars rose from 760 in 2010 to 70.000 in 2014 and that the choice of models increased from three to 30 in the same period [12]. This is due to the fact that the mentioned ratio is increasing for specific use cases such as in-city transportation.

The integration of EVs in the power grid requires changes to the power infrastructure and also to the networks and systems that are responsible for handling smart grid data.

2.3.4 New Business Models

A smart grid enables the collection of information about the state of the electrical infrastructure that was the power grid and electricity customers, which was previously too costly or, in some cases, impossible to obtain. This information has different value for different electricity system players. For instance, electricity suppliers require clients' consumption data to bill and to offer energy services to the end customers. In 2013, the Smart Grid Task Force (SGTF) Expert Group 3 (EG3) published a report about the options on handling smart grids data. In this report, EG3 included the following three different case models [13].

- DSO as market facilitator - The DSO provides the operational data and market facilitating data (e.g., data about customers, their technical possibilities and their

consumption or production) to the remaining players through a data hub that is within its ownership and control;

- Third party market facilitator - Independent Central Data Hub (CDH) - The data is available to the authorized players through an independent central communication platform based on one or several data hubs;
- Data Access-Point Manager (DAM) - A certified company provides access to the data and functionalities of the field devices to any certified player and/or consumer/prosumer, using an implemented communications network.

In the three cases, there are differences on who processes, communicates and stores the data and also on the connections between the different electricity system players. Depending on which is their specific case and especially on the first case, DSOs must apply adequate security controls to protect the information and their costumers.

2.4 Smart Distribution Grids

”A smart grid is an electricity network that can cost efficiently integrate the behavior and actions of all users connected to it - generators, consumers and those that do both - in order to ensure economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety”[14]. It is envisioned by the EC and the electricity system players as the answer to the new challenges presented in Section 2.3.

The smart grid is the current evolutionary step from the traditional power grid, meaning a change from a locally and manually controlled electrical network, which was completely isolated from any kind of computers or communications network, to a remotely controlled and automated cyber-physical infrastructure. It is a highly complex infrastructure with many different systems, components and connections between them and even to external infrastructures. New smart grid functionalities and use cases are still being developed, which call for constant R&D of new systems and technologies.

There are some systems, components, functionalities and use cases that could be considered core elements of the smart grid concept. A description of the most relevant aspects, in regard to our work, is given below.

2.4.1 Facilities and Cabinets

A smart grid’s assets are distributed through a wide geographical area, which can often reach a country-wide dimension. They are commonly hosted inside facilities and cabinets, which provide access to essential utility services, such as electricity and communications supply, and physical protection from a variety of natural and human-origin risks.

- Control room - The physical location from where the DSO's teams manage, operate and monitor the electrical network, communications network and systems in a resourceful and controlled environment. The electrical network control room or command center may be separated from the remaining functions. There are usually redundant and fallback facilities. All control rooms are redundantly connected to the data centers.
- Data center - The physical location where the central systems are hosted with the necessary physical security, utility and climatization services. There are usually redundant facilities.
- Primary Substation - A facility that hosts one or more HV to MV transformers together with the necessary electrical equipment and local systems. A primary substation count can reach the hundreds.
- Secondary Substation - A facility that hosts one or more MV to LV transformers together with the necessary electrical equipment and local systems. A secondary substation count can reach the tens of thousands.
- Automation cabinet - The local systems of pole-mounted secondary substations and DA devices are hosted in cabinets. A DA device count can reach the thousands.
- Consumer/prosumer house - Although they do not count as smart grid assets, the consumer/prosumer houses host the smart meters and MG equipment. Their number can reach the millions.

2.4.2 Electrical Network

The primary substations connect electrically to the transmission network, to the DG, to the secondary substations and to a subset of other primary substations for redundancy purposes. The secondary substations and the DA devices are installed along the primary substation's MV feeders. The secondary substations connect to the primary substations, to the consumers/prosumers and to each other, also for redundancy purposes. Typical electrical network components include transformers, lines, busbars and switchgear. Auxiliary power systems and power factor correction batteries can also be present.

- Transformer - Transforms a higher voltage to a lower voltage (i.e., HV to MV or MV to LV) or vice-versa, depending on the current flow.
- Switchgear - Enable the electrical isolation and/or protection and/or operation of the electrical network and loads, such as motors, heaters, lighting and capacitors. Disconnect switches, circuit breakers¹ and fuses are examples of switchgear.

¹As defined in the IEC 60947-2 standard, a circuit breaker "is capable of making, carrying and breaking

- Line - Allows the flow of current between the loads that are electrically connected in both of its ends.
- Busbar - Allows the flow of current between the lines that are connected to it.

2.4.3 Sensors and Actuators

Sensors and actuators are the interface from and to the electrical reality, respectively.

- Sensors - Detect and make measurements, such as electrical quantities (e.g., currents and voltages) and temperatures. The power grid is comprised of many different equipment and components that are exposed to many different physical threats; e.g., climatic conditions and deterioration from use. There are many different types of sensors (and others are being developed) to prevent and detect power grid faults and failures. Example applications include detection of hot spots in power equipment, open door, flood and fire. These sensors are intelligent, which means that they are able to monitor a given equipment but they also monitor themselves to send alarms about their own health and performance. This enables operators to distinguish between situations when the monitored equipment is fine and situations when the sensor is damaged. Some information is valuable to the grid operation and it can be integrated in the SCADA system. The other can be valuable to management or maintenance and can be integrated in business analytics or monitoring systems. Instrument transformers and protection relays are examples of sensors.
- Actuators - Interact with the electrical reality upon request or as automatic response to predefined events. Disconnect switches and circuit breakers are examples of actuators.

2.4.4 SCADA and DMS

The SCADA system and the Distribution Management System (DMS) are the interface between the command center and the local systems. They are central systems located at the DSO's data center, where they are typically connected with other central systems such as the Outage Management System (OMS)², the Geographic Information System (GIS)³ and the Historical Information Manager (HIM)⁴.

currents under normal circuit conditions and also making, carrying for a specified time and breaking currents under specified abnormal circuit conditions such as those of short-circuit" [15].

²The OMS maps certain SCADA events to power outages and provides a graphical environment for the grid operators to manage power outage incidents.

³The GIS contains and provides information about the electrical network assets, including what and where they are.

⁴The HIM stores electrical network information older than a given age for consultation

- SCADA servers - Retrieves telemetry and sends operators' commands to field equipment, such as primary substations, secondary substations, sectionalizers and reclosers.
- DMS servers - Maps the information collected by the SCADA with the geographic information of the GIS and provides advanced functionalities to support operators' decisions, such as fault location and power flow calculation.
- Workstation - Provides a visual environment (i.e., a Graphical User Interface (GUI)) for the grid operators to manage and control the electrical network.

Local systems provide monitoring and control of sensors and actuators in primary substations, secondary substations and feeder automation devices, handling internal and external communications, local and remote access and automation functionalities. They consist of one to several devices, depending on the specific solution.

The SCADA systems connect also to Network Time Protocol (NTP) servers (e.g., Global Positioning System (GPS)), centrally and locally, to register the correct timing of events, which is critical to a proper analysis.

2.4.5 Substation Automation

Substation automation consists of automating a given set primary substation behaviors to safeguard the costumers, their infrastructures, the DSO's workers and the power grid. Examples of automation use cases are voltage or frequency shedding and feeder reconections after fault occurrences. Inside the substation, there is typically a local SCADA system with the following components:

- Central Unit (CU) or Substation Controller (SC) - Connects through the internal Local Area Network (LAN) to a set of Intelligent Electronic Devices (IEDs), to which it sends commands and from which it receives telemetry information and events. These commands can be either dispatch and control center SCADA commands that it forwards from the external primary substations' Wide Area Network (WAN) or automation commands that it generates based on its internal processes. Alternative primary substation internal network solutions include physical connection of the IEDs directly to the SC, without an internal LAN.
- IED - Monitors the sensors and controls the actuators, such as the electrical protections, based on the commands received from the SC or locally given through an integrated user interface. They communicate with each other for providing advanced functions such as protections coordination.
- Human-Machine Interface (HMI) - Connects to the SC to provide a visual environment (i.e., a GUI) for the local control of the electrical network.

2.4.6 Distribution Automation

DA is automation of the smart grid at the MV level, including secondary substations and feeder automation devices. The Remote Terminal Unit (RTU) is typically the main component in secondary substations and feeder automation devices.

- RTU - Provides SCADA communications and automation functionalities.

Developments in this area are quite recent, when compared with substation automation, and examples of use cases are self-healing, load balancing and Volt/VAR control. It is different from what existed before by introducing communications between field equipment at different facilities; for example, in self-healing (see Section 2.6), the RTUs's might communicate between themselves and/or with the primary substation's SC.

2.4.7 Advanced Metering Infrastructure

When looking into the history of AMIs, we find that first there were Advanced Meter Readings (AMR) infrastructures, serving the purpose of remotely collecting meter readings at will or by schedule, from a given set of meters. Then, smart metering emerged with remote meter readings as its core functionality and a whole set of new functionalities and use cases, such as registering electricity consumption data up to every, e.g., 15 minutes and enabling access to this information to the customer. The customer can have a better understanding of its own consumption habits and is able to change them [14]. If a secondary substation is integrated in a smart metering solution, there is also a Data Concentrator (DC) handling communications between the central systems and the smart meters at customers houses.

- Smart meter - Installed at the customers' house, it registers the consumption readings, at a configurable frequency, and sends them upstream to the DC. Some smart meters can connect with other devices in a Home Area Network (HAN), providing an HMI that can be used by customers to see their data. They can also have circuit breaker capability to isolate the house from the electrical network, which can be activated at central systems command.
- DC - Installed at the secondary substation, it transmits the consumption readings to the central systems. EDP Distribuição⁵ uses its own unique solution. It uses only one device - the Distribution Transformer Controller (DTC) - that combines DA and smart metering capabilities. This approach gives the capability to correlate DA and smart metering information locally, therefore potentiating future smart grid use cases.

⁵EDP Distribuição is the largest Portuguese DSO with little under six million customers. It is a regulated company belonging to the EDP Group.

In EC Member States where the DSO manages the smart metering infrastructure, the data is then transmitted to the corresponding Suppliers. An AMI also enables the management of meters and concentrators.

2.4.8 External Connections

The Industrial Control Systems (ICS) network, containing the smart grid's systems and communications, is segmented and segregated from external networks, such as the DSO's business network or the Internet, which is not a complete isolation due to the fact that some smart grid's systems and functionalities may require the processing of external information, external maintenance and updates. Therefore, there may be external connections and communications to, for example, the TSO, DG companies, external service providers and manufacturers' update services.

2.4.9 Communications

Smart grid communications use a wide range of different technologies, means and protocols. Here we describe a few demonstrative but non-extensive examples.

- Central systems - Typically use IP-based protocols in 100 Mbps, 1 Gbps and 10 Gbps ethernet networks.
- Central systems and primary substations - Use a mix of legacy proprietary and standard-based protocols and technologies, such as IEC 60870-5-101 and IEC 60870-5-104, in Plesiochronous Digital Hierarchy (PDH), Synchronous Digital Hierarchy (SDH), Internet Protocol (IP)/Multiprotocol Label Switching (MPLS) and General Packet Radio Service (GPRS) networks, through optical fibers, microwave radio links and mobile technologies.
- Primary substation LAN - Uses proprietary and standard-based protocols, such as IEC 61850, through ethernet and optical fiber networks.
- Central systems and secondary substations - Typically use standard-based protocols, such as IEC 60870-5-101 and IEC 60870-5-104, in GPRS and Global System for Mobile Communications (GSM) networks, through mobile technologies. Secondary substations connecting to priority costumers or with a high number of transformers (and the corresponding local systems) can be connected through optical fibers.
- Secondary substations and smart meters - Use standard-based protocols, such as Device Language Message Specification (DLMS)/Companion Specification for Energy Metering (COSEM) through Power Line Communication (PLC). Some smart meters can connect directly to the central systems through GPRS mobile networks.

2.4.10 Microgrids

A microgrid is a local electrical network that comprises enough energy resources (e.g., generation, storage, distribution and supply capabilities) to operate without being electrically connected to the main electrical network. Its mode of operation can also be called island mode. The possible connection to the main grid is regarded as a contingency solution.

2.5 Smart Grid Concerns

The increasing number of highly connected smart grid systems and components, of new types and with increasing relevance, complexity and dependence on external services and information, expands and amplifies the smart grid's surface of exposure to new types of threats and vulnerabilities. For example, the electrical network is exposed to "traditional" threats, such as: collision of tree branches, thunder storms, rain, strong winds, fires, animals, construction work-related accidents, programmed interventions and vandalism, which are common causes for electricity outages [16] and that are also threats to the communications network. The systems and communications are exposed to non-malicious faults (e.g., software faults), and to malicious faults (e.g., cyber attacks). Threats such as hacktivism have now turned their attention to energy infrastructures; reports such as ICS-CERT Monitor show an increasing trend in the number of security incidents reported by the energy sector [17, 18]. In addition, cyber security incidents such as Stuxnet reveal that the complexity of attacks is increasing [19, 20].

In the second half of 2015, three Ukrainian electricity distribution companies were targeted by successful cyber attacks to their business and ICS networks, including unauthorized access to the operation workstations and to substation equipment. It is also relevant to note that the attacks to the ICS network used information - user credentials - obtained through attacks to the business network, exploiting the existing connections and dependencies between the two networks. These attacks exploited vulnerabilities related with security design, monitoring and awareness, resulting in several electricity outages that affected 225.000 costumers across several areas and lasted for several hours on December 23, 2015. They gain further relevance due to the fact that they "are the first publicly acknowledged incidents to result in power outages" [21]. Non-malicious faults or successful attacks may have consequences in the performance, privacy, security and dependability of the smart grid infrastructure and/or information and to those who depend on them, including consequences to persons' lives.

2.5.1 Performance

Smart grid functionalities support different DSO processes, which have different performance requirements. Distribution system operation and maintenance requires real-time information about the electrical network state, including electrical measurements and the state of protection devices. Network and asset management require historical information about the grid operation and device behaviors for network planning and fault prevention. There are also different requirements to real-time communications. For instance, the situation when an operator sends a command to a protection device inside a substation and waits for a confirmation is different from the situation when a maintenance worker accidentally touches an exposed electrified line, triggering a protection device. Therefore, substation automation protocols (e.g., IEC 61850 GOOSE) are designed to be faster than SCADA protocols (e.g., IEC 60870-5-101 or IEC 60870-5-104).

2.5.2 Privacy

The smart grid handles sensitive information of different nature, which is relevant, for example, to its management, operation and maintenance, to the market processes and to the costumers. This information must be identified and classified in compliance with the DSO's information classification policy and it must be protected in compliance with the DSO's information security policy and applicable legislation and regulations. Personal data, which is "any information relating to an identified or identifiable natural person", as defined in the new European General Data Protection Regulation (GDPR) [22], has been a particular focus of concern in regard to smart metering. A main issue is that the collected electricity consumption data might expose the costumers' life behavior, endangering his/hers security. The Opinion 12/2011 on smart metering of the Article 29 Data Protection Working Party demonstrates that "personal data is being processed by the meters, so data protection laws apply" [23]. Between 2012 and 2014, the SGTF Expert Group 2 (EG2) developed a Data Protection Impact Assessment (DPIA) template for smart grid and smart metering systems to guide the "organizations who initiate or already manage smart grid deployments as well as those introducing changes to existing smart grid architecture platforms in identifying and assessing the privacy risks of these initiatives" [24].

2.5.3 Security

Smart grid information goes beyond the aforementioned electrical network telemetry data and the electricity consumption data, including also: historian data, GIS data, network topology information, the organization's policies, procedures and low-level instructions, manual and automated logs, reports, change management information, equipment configurations, backups, working files and user credentials, among others. This information

must be available when required, without illegitimate and/or undetected modifications, to those who need it and that are authorized to access it. The same applies to other smart grid assets, besides information, such as facilities, systems and components, services and people. If there is a confidentiality breach and confidential information falls in malicious hands, it might further compromise the smart grid infrastructure, the organization and the costumers' privacy. If the information was illegitimately modified and that goes undetected, the control center may take wrong decisions that might further compromise the smart grid infrastructure and the supported processes. If a smart grid component is behaving erroneously or maliciously, the control centers' commands or the automation actions might have different results from what was expected. If at least one smart grid asset is unavailable when required, there might be an electricity outage and/or the control center may not have the resources that it requires to detect and handle faults and attacks. Other possible consequences related with the loss of confidentiality, integrity and/or availability are loss of reputation and financial loss. Regarding the reputation loss, an organization might be regarded as one which did not take the appropriate care of the infrastructure or of its costumers data, being untrustworthy and/or unreliable.

2.5.4 Dependability

The smart grid provides internal services that support the internal DSO's processes, such as the monitor and control of the electrical network. As smart grid functionalities become increasingly more integrated in the DSO's processes, the control center becomes more dependent on its systems and communications. The SCADA system has become "their eyes and hands" to the field. The smart grid provides also external services that support its business processes, such as the distribution of electricity and the provision of data to the market. The power grid is a critical infrastructure that, in case of incapacitation or destruction, would have a debilitating effect on the economy and public safety [25, 26]. This is due to the fact that many things in our world depend on electricity to operate, including other critical infrastructures, such as hospitals, ports, airports, communications, finance, water and wastewater systems. It is in fact a two-way relationship, given that the smart grid may rely on external infrastructures with regard to its communications. Moreover, the existing connectivity between the critical infrastructures and corresponding ICS networks might be exploited by the adversary to attack a critical infrastructure with another critical infrastructure; e.g., causing electricity outages to the health, transport or communications sectors or compromising the smart grid communications from within an external communications network. This "opens the doors" for the use of the smart grid as a weapon in wars and similar conflicts.

2.6 Automation and Self-healing Approaches

Substation automation and DA have the capability to handle faults occurring in the electrical network, automating incident response to different lengths of the FLIR sequence for different voltage sections. This self-healing behavior is the current scope of a set of world-wide real-world deployments and R&D initiatives, mainly focusing in its functional aspects.

Self-healing is an emergent behavior that results from the network having intelligence to prevent, detect, locate, isolate and recover from electrical faults by itself or, as an alternative, to propose to the operators the recovery plan - a sequence of maneuvers that can be validated by the dispatch and command centers and then executed by them or by the grid itself. Its purpose is to enable a faster and more effective response to power outages, thereby reducing the number of costumers affected by a fault, the number of unenergized costumers after fault isolation and recovery, the outage time for affected costumers and the impact of isolation and recovery maneuvers in the network.

There are distinct self-healing approaches in also different maturity states, ranging from real-world deployments to R&D. There is a trend towards intelligent communicating self-healing solutions that use local controllers at primary and/or secondary substations. However, non-communicating solutions such as time-based coordination and selective coordination are already deployed or being deployed in large geographical areas and, depending on the specific type of employed equipment, they provide a self-healing solution with a performance comparable to that provided by intelligent solutions, although not as flexible as the previous. Both solutions coexist at the present moment and will continue to do so in the future. Moreover, the decentralization of the decisions and control from the command center to the local controllers increases the resilience of the solutions to network partitions that isolate a substation, a group of substations or the central systems.

The following reference examples aim at providing an overview at the different types of approaches to provide a better understanding of the use case.

2.6.1 The EDP Distribuição Case

EDP Distribuição is currently following five different automation and self-healing approaches, at different life-cycle states.

1. A first approach is based on substation automation capabilities. It is comprised only of the protection device that is installed at a primary substation. When there is a fault in a MV feeder line, the protection opens, which is followed by a given number of reconnections to get past any fugitive faults; e.g., a tree branch that touches the overhead distribution lines for an instant, causing a short-circuit fault that triggers the electrical protections. If the fault is fugitive, then the network is healed and the electricity flow is consequently restored. Otherwise, all the costumers supplied by

the faulty line stay unenergized until the dispatch and command centers remotely reconfigure the network, which is only possible in mesh or ring networks, or until there is a local intervention.

2. A second approach comprises the installation of sectionalizers with voltage presence detection and fault current detection capabilities in overhead MV feeder lines. These sectionalizers can be remotely controlled to disconnect the feeder where they are installed. They also have an automatism that makes them open when voltage absence is detected, close when voltage is detected and lock themselves open after a given number of reconnection cycles. In an automated feeder, there is at least one sectionalizer installed downstream from the protection device that is at the primary substation. When there is more than one sectionalizer, their parametrization must be coordinated so that, when there is a fault, the only sectionalizer that locks open is the one immediately before the location where the fault occurred. All the other sectionalizers stay closed, therefore isolating the fault and assuring electricity supply in the sections of the feeder that are healthy. This coordination is accomplished at electrical level and by configuring adequate opening, closing, maneuver and locking times at each sectionalizer. More than one reconnection cycle may occur during a reconnection process, which are visible to the costumers that are being supplied by the faulty feeder; e.g., lights may flicker during a couple of seconds until they finally stay on or off. Therefore, this approach increases the number of energized costumers after a fault. However, the number of reconnection maneuvers can be significant and noticeable by the customers. The sectionalizer is also capable of sending alarms to the SCADA systems when a fault current is detected. This approach is already in a roll-out state.
3. A third approach comprises the deployment of reclosers with protection capabilities in overhead MV lines. These devices act similarly to the protections that were described in the first approach but they are installed along the feeder lines like in the second approach. Moreover, when in a normally open configuration, they can detect the absence of voltage downstream and close automatically to energize the unenergized section. If there is a fault in that section, the protection trips and the recloser returns to its previous open state. Therefore, this approach increases the number of energized costumers after a fault, when compared with the first approach, while reducing the number of maneuvers and visibility to the costumer of the second approach. This approach has been tested with good results [27].
4. A fourth approach comprises a DMS fault location functionality. It uses measured fault information from a primary substation and GIS information to calculate the possible location of an electrical fault and the affected MV and HV sections of the electrical network. The result is provided to the electrical network operators, allow-

ing them to position repair teams on the field with increased efficiency, decreasing fault repair times and outage times. When a fault occurs in the MV network and it triggers a primary substation protection device (see the Approach 1), the central unit reports the measured fault resistance and reactance to the central systems. A DMS process uses these values, together with a user configurable calculation error, to calculate the fault impedance and the maximum and minimum distances between the triggered protection device and the probable fault location. The algorithm assumes the fault impedance to be the same as the network impedance associated with the lines between the device and the real location, minus the error. Therefore, it uses the network topology to identify all line sections ahead of the device that are located within the distance interval as possible fault locations. The considered topology must reflect the stable network state before the fault occurrence. To find the right moment from when to extract the topology, the algorithm assumes the transitory states to have occurred within a limited time frame around the instant when the fault measurements were received, the length of which is user configurable. In the future, information from fault detectors and other devices (see the Approach 2) located in the calculated sections or in contiguous sections will be used to further narrow the possibilities. The reliability of the results depends on the correct characterization of the electrical network equipment. If the line impedances are incorrect, the algorithm might provide bad results that might delay, instead of accelerating, the work of the operators and repair teams [28].

5. A fifth approach comprises the use of distribution intelligence at the primary substations together with the sectionalizers that are used in the second approach. The sectionalizers send fault alarms to a Smart Substation Controller (SSC) at a given primary substation when a fault occurs. Based on the received information, the SSC has the capability to determine which devices must stay open or closed to isolate the faulty feeder section and energize the sections of the feeder that are healthy. This approach uses a RF Mesh communications solution for the communications between the sectionalizers and the SSC. If, for some reason, the primary substation is not accessible through RF Mesh, communications can go through a second primary substation that acts as a communications network redundancy and routes the information to the first primary substation. The recovery plan is presented to the dispatch and command centers for validation and approval. If it is approved, the SSC executes the plan by sending commands to the sectionalizers. This solution allows the reduction of the number of maneuvers in the second approach type of sectionalizers. It is being tested at the Portuguese city of Batalha [29, 30].

2.6.2 The Stedin Case

Stedin⁶ operates an underground network that does not allow the installation and operation of automatic reclosers. Therefore, their self-healing solutions are based on primary and secondary substations only.

1. A fully decentralized approach is deployed in the city center of Rotterdam [31, 32]. In this approach, intelligence is distributed between a primary substation and the secondary substations that are connected to two of its MV feeders, which are part of a ring with a normally open point. The primary substation is equipped with protection relays, a SCADA RTU and a self-healing RTU. The secondary substations are equipped with RTUs, Fault Passage Indicators (FPIs) and voltage presence detection capability. In the case of a fault occurrence, each RTU executes a software restoration routine that handles the communications with the previous and the next secondary substations RTUs to execute FLIR steps automatically. The RTUs pass messages between them, according to their order in the feeder relative to the primary substation protection that tripped until the faulty section of the feeder or secondary substation has been isolated and the remaining sections have been energized. A telecom provider's GPRS private APN is used for the communications between the RTUs.
2. A second self-healing project in the Rotterdam harbor district comprises the installation of a regional controller in a primary substation and local control units in each downstream secondary substation. The regional controller ensures automatic FLIR, interface to Stedin's control center and hosts the regional, centralized self-healing applications [33, 34].

2.6.3 The SEGRID Project Case

The fifth use case of the SEGRID European Project is Automatic reconfiguration of the power grid, which has as goal the location and isolation of a fault through network reconfiguration, while minimizing the number of affected costumers and power parameters [35]. The use case description refers to three different scenarios.

1. In the first scenario, isolation and restoration of faults in the MV network is decided centrally. A RTU at a primary substation collects information about the electrical network topology and equipment parameters from the central systems. In the case of a fault occurrence, it uses the afore mentioned information, together with information that it collects from the downstream secondary substations and automatic reclosers, to locate the fault. The isolation may be controlled by the primary RTU itself, by sending commands to the secondary substations and reclosers, or by the

⁶Stedin is one of the largest Dutch DSOs with more than two million customers.

grid dispatch center. The recovery is controlled by the RTU itself, while assuring that the reconnected load stays within the capability of the newly reconfigured network.

2. In the second scenario, isolation and restoration of faults in the MV network is distributed by secondary substation's RTUs, which communicate with each other through peer-to-peer communications to decide the topology of the MV network. Any topology change requires an update to the secondary RTUs topology models. Isolation is performed by the secondary RTUs themselves. For restoration, it is also necessary to assure that the reconnected load stays within the capability of the newly reconfigured network.
3. A third scenario refers to the minimization of losses in the MV network using switching. The goal is to optimize the network to prevent faults from happening. As in the first scenario, here the primary RTU also collects information about network topology and equipment parameters from the central systems. This information is used together with information collected from the electrical network, such as electrical measurements and the state of devices, to forecast also short term future needs and to calculate an optimal network configuration. This configuration can either be proposed to the dispatch center or executed automatically.

2.6.4 The e-balance Project Case

The e-balance European Project refers to five fault detection, location, isolation and restoration use cases, one of which includes a fault forecasting capability that is not mentioned in any of the above [36, 37, 38, 39].

1. The use case numbered 22 is fault detection and location in LV networks. A grid management unit is installed in the secondary substations, which combines network topology information with fault related information that is retrieved/received from downstream devices. Sensors at LV distribution cabinets send alarms when configured current or voltage thresholds are reached. They can also be polled by the management unit to confirm persistent faults. RF MESH smart meters send "last gasp" messages upon severe fault occurrences. PLC PRIME meters can be polled by the management unit, to which they will only respond if they are placed in a healthy feeder. Communications between the sensors/smart meters and the management unit use the DLMS protocol. The management unit identifies the faulty phase and feeder segment and communicates this information to the MV grid management unit at the upstream primary substation.
2. The use case numbered 23 is fault and fused luminaries detection and location in public lighting. The LV grid management unit uses topology information, namely

the number of luminaries and impedance of each, together with current and voltage measurements, which are retrieved from sensors deployed at each phase along the public lighting feeders. When a feeder section is healthy, its current and voltage measurements are used to calculate the reference impedance of that section. Changes in the calculated impedance of a feeder section may imply a faulty line or fused luminary. The topology information is used to forecast the number of faulty luminary in each faulty segment. Communications between the sensors and the management unit use the DLMS protocol.

3. The use case numbered 24 is fault prevention in LV networks. The LV grid management unit prevents the network from reaching limit operation conditions, namely, it prevents voltage limit violations, caused by micro producers, and thermal limit violations, in secondary substations and distribution cabinets' protective fuses. LV sensors send alarms to the management unit when a current, voltage or thermal threshold is surpassed. They can also be configured to send measurements on a given schedule or be polled by the management unit. Smart meters at costumers' houses can be polled by the management unit to send voltage measurements. They also act as a gateway for communications between the LV grid management unit and a Distributed Energy Resource (DER) grid management unit, which is itself an interface for communications with the MG Photovoltaic (PV) inverter. By correlating received and retrieved information, the LV grid management unit detects effective and imminent voltage violations and calculates control set-points that are sent as commands to the DER grid management unit to control the PV inverter, thereby controlling the power injection. The DER grid management unit can also calculate provisional set-points, upon detecting voltage violations. When the LV grid management unit detects current or thermal violations, it sends load curtailment commands to the smart meters of non-priority costumers. The LV grid management unit reports the afore mentioned occurrences to the MV grid management unit. Communications between the smart meters and the LV grid management unit use the DLMS protocol. Communications between the LV grid management unit and the DER grid management unit use the MODBUS protocol.
4. The use case numbered 29 is fault detection and location in MV networks. As already mentioned, the MV grid management unit is installed in the primary substations. It combines network topology information with fault information that is requested/received from downstream devices such as MV fault sensors at secondary substations, automatic reclosers and primary substations' protections. After a fault occurrence, the management unit waits for the passing of a configurable timeout, during which the network is trying to heal itself following an approach that is similar to EDP Distribuição's second or third approaches. If it is a persis-

tent fault, then the management unit will calculate the location of the faulty feeder segment or equipment. It then sends this information to a top level grid management unit. Downstream devices communicate with the MV grid management unit through standard IEC protocols.

5. The use case numbered 30 is automatic grid service restoration - self-healing - in MV networks. After the fault has been detected and located (see e-balance's use case 29 description), the MV grid management unit will use this information together with network topology information to plan a set of reconfiguration actions to isolate the fault and restore electricity to a maximum of affected costumers. This is achieved by transferring loads between MV feeders, without reaching network limit operation conditions (see e-balance's use case 24 description). Similarly to EDP Distribuição's fourth approach, the plan is presented to the electrical network operators for validation. If it is accepted, the MV grid management unit will send commands to reconfigure the network. This use case can also be executed in automatic mode, through which the human validation step is not performed.

2.7 Cyber Security and Dependability Controls

The faults in systems and communications are handled with the support of commercially available solutions, which provide perimeter and in-depth protection capabilities. The majority are Commercial off-the-shelf (COTS) solutions that were designed for general Information and Communication Technologies (ICT) use and not specifically for Operation Technology. As mentioned in Sections 2.4 and 2.5, the smart grid has specific equipment and requirements that impose restrictions to the solutions that can be used in the ICS network. Moreover, in the central systems and from one substation to the other we may find different generation deployments (e.g., legacy equipment, different network topologies and/or lesser communication capabilities), which prevents the use of "one size fits all" solutions. These solutions are often based on a predefined set of rules, signatures or use cases, which automate fault handling for a set of specific cases. Section 2.7.2 presents a set of relevant R&D works that focus on smart grid specific solutions. Report [40] goes further to identify a set of gaps in the current solutions and discuss how R&D is trying to fill the identified gaps. An organization should select and setup the security solutions that best apply to its specific context at a given moment but also thinking about the future. For instance, it should also setup the process and reserve the necessary Human Resources (HR), including knowledge and experience, to operate them correctly. Also, security solutions should be integrated in a security platform in such a way that any given one complements the others and all of them together protect the organization and its processes from the threats.

2.7.1 Currently Used Solutions

The following are examples of security and dependability solutions that are currently used in smart grids. For detailed explanations on the workings of the mentioned technologies, refer to [41], [42], [43], [44], [45] and [46].

Physical Security and Safety

Most smart grid assets exist in the physical space or have some sort of support, including people, systems and information, which must be protected from physical threats, such as extreme weather, fire and intrusions. Physical barriers (e.g., walls, doors and windows), human guards, master key, card or electronic key-based access control, Closed-Circuit Television (CCTV), fire detection and prevention and climatization systems are examples of common controls that can be found in the smart grid facilities.

Perimeter Protection

Perimeter protection can be seen as the first layer of an organization's cyber security platform. It isolates the organization's network from the outside and provides protection against external threats. Network and host-based firewalls are a basic building block of perimeter protection solutions in smart grids, allowing or blocking communications based on a set of configured rules.

Network Segmentation and Segregation

An entity "having access to the network" should be a relative concept. The network must be segmented and segregated so that, e.g., the access to one host at the data center does not mean also access to the substations. It must take into account the roles of the network devices and hosts and the roles of the users. Role-based access control (RBAC), firewalls, Virtual Local Area Networks (VLANs) and Virtual Routing and Forwarding (VRFs) are examples of used controls.

Virtualization

Virtual Machines (VMs) can be used to partition a physical machine's resources by several guest Operating Systems (OSs) and the corresponding applications, while providing fault and security isolation at the hardware level [47]. The decision to virtualize a machine must take into account the type of resources required by the installed applications (e.g., I/O requirements) and how these are used.

Hardening

Systems and devices can be hardened through disabling unessential applications, processes, services, ports and enabling security enhancing configurations.

User and Access Management

The smart grid users can be grouped and assigned access profiles according to their roles in the management, operation, maintenance and support of the different systems and components in the different smart grid layers. There is centralized management for the computers and devices that support it, which should be a basic requirement for new acquisitions. There are emergency local users for when the centralized authentication and authorization service is unavailable. Emergency local users, centralized directory, Authentication, Authorization and Accounting (AAA), Privileged Access Management (PAM) and Public Key Infrastructure (PKI) are examples of used controls.

Secure Communications

The use of secure protocols (e.g., Hyper Text Transfer Protocol Secure (HTTPS) and Secure Shell (SSH)) is preferred over any other alternative, if the computers and devices support it, which should be a basic requirement in the specifications for new acquisitions. The external access to the ICS network uses a Virtual Private Networks (VPNs), which is used for remote interventions and by external service providers. Secure protocols, VPNs and cryptography management are examples of common controls in smart grids.

Intrusion Detection and Prevention

Network and host-based IDS and Intrusion Prevention System (IPS) can be used to detect and prevent suspicious communications in the smart grid. They can employ a set of predefined signatures or they can make comparisons with a predetermined usage pattern. Nevertheless, due to the false positives that are typically associated with these solutions, the prevention capability should be used with extra caution if it is applied to communications related with the critical processes.

Anti-malware Protection

Firewalls, IDS, whitelisting and antivirus are examples of controls that are used to protect the smart grid from malware. The host-based solutions require deep knowledge about the applications, processes and services that are executing at each machine. New rules and signatures must be tested for their impact in the performance of a representative set of hosts before they are accepted into production. This requires the commitment of the DSO and the solution support providers to work closely together during the testing and also to decide how to proceed if the new rules or signatures have a negative impact (e.g., an

exception may be added or the application may be changed). They also require knowledge about the hardware capabilities of each machine, as legacy equipment may support only a combination of the aforementioned solutions by specification or due to their impact in performance. If used incorrectly, malware protection may degrade or even stop critical processes.

Dependability Techniques

Redundancy is the most common dependability technique used in smart grids to protect against non-malicious crash faults. It comes from a time when the architecture and systems were simpler, the ICS network was more isolated from any external communication networks and the cyber attacks to electrical infrastructures were not the concern they are today. There are usually redundant servers, workstations, communication channels and electricity supplies to distinct levels in the different ICS network segments and smart grid facilities. Diversity appears as a consequence of the technological renewal and evolution of the smart grid, which becomes more evident in Operation Technology technology and especially in the substations, as they have longer life cycles.

Security Information and Event Management

In the smart grid, systems and network devices generate log files such as syslog [48] where they register security related events such as user logins and logouts. SIEM systems can be used to collect the logs, correlate the security events and send alarming messages, based on a set of configured use cases.

System and Network Management

It is common for systems and network devices to be able to report on their performance and health state through Simple Network Management Protocol (SNMP) traps. The most current version of this protocol is SNMPv3 [49]. This information can be consolidated and managed in a central management system, such as a Network Management System (NMS), which can send alarming messages, based on a set of configured policies.

Backup Management

Configuration and data backups for smart grid assets are performed locally or remotely to on-site and/or off-site locations. The data retention periods comply with the functional requirements, organizations policies and procedures and applicable regulations and legislation.

Security Update Management

As with IDS and antivirus updates, other security patches and updates, such as those applied to the OS, are also tested for compatibility with the critical applications before they are accepted into production. They also require the DSO and the solution support providers to work closely together during the testing and to decide how to proceed if the changes have a negative impact. Solutions that are close to the end of support or End-of-life (EOL) should be replaced by solutions with the required level of support.

Guides, Standards and Certification

There are guides and standards that can be used by the DSO to identify the assets that require protection, to perform threat, vulnerability and risk assessments and to identify the necessary security and dependability controls. The DPIA template mentioned in Section 2.5.2, the ISO/IEC 27000 standard series - Information security - and ISO/IEC 31000 - Risk management are examples of such documents. The DSO may go even further to certify one or more of its core processes. In this regard, the ISO/IEC 27001:2013 provides requirements for the implementation of a certifiable Information Security Management System (ISMS).

Audits and Penetration Testing

It is important for the DSO to perform regular internal and/or external audits to review the security and dependability implementation in comparison with any adopted standard and the company's policies and requirements. Complementing the documentation analysis, the audits should also include penetration tests to the robustness of the implementation. Penetration testing can also be used to identify security vulnerabilities in the deployed smart grid infrastructure and in systems and components before they are accepted into production.

Training and Exercises

User training and awareness and incident response exercises are essential to the correct operation and maintenance of the smart grid infrastructure on a daily basis and in response to unexpected events and incidents.

2.7.2 Relevant Research and Development

Smart grid cyber security has been gradually gaining interest for R&D work in Europe, being a current and recurring topic in European projects and initiatives, conferences and publications. The JRC publishes an outlook report on European smart grid projects since

2011, with Smart Grid Projects Outlook 2014 being the most recent report [50]. The majority of the reported projects focus mainly on the challenges of Section 2.3 and related issues, namely, smart network management, integration of DER, integration of large scale RES, aggregation, smart customer & smart home, electric vehicles and smart metering, with some mentioning privacy and/or cyber security as requirements of the proposed solutions. More recently, the European projects SEGRID and SPARKS compiled a list of privacy, security and dependability focused European projects, such as themselves, which focus on the R&D of policy recommendations, new tools and technologies and improvement of existing ones in the aforementioned areas [40, 51]. The following are relevant examples to our work.

e-balance

The e-balance project "aims at integrating the energy customers into the future smart-grids in order to address future environmental problems with holistic technical solutions based on ICT, new business models and citizens' behavior under real world conditions" [52]. It follows a security-by-design approach in the development of the proposed solutions, which are focused on a set of use cases that include FLIR and self-healing.

HEAT

The HEAT project "will examine new design and implementation techniques for homomorphic cryptography, as well as a thorough security analysis" [53]. One of its objectives is to demonstrate the applicability of Somewhat Homomorphic Encryption (SHE) to smart grids, which has application in the processing of privacy sensitive smart metering data by the DSOs for smart grid management and operation functionalities.

LV-Pri20

The LV-Pri20 project will develop an automatic verification software for the formal and automatic analysis of privacy-preservation in ICT [54]. Its use cases include smart grid applications included, such as the Open Smart Grid Protocol [55].

SEGRID

The SEGRID project main objective is "to enhance the protection of smart grids against cyber attacks" [56], by focusing on an in-project defined smart grid reference model and a set of future smart grid use cases, which are based on previous EC published work. A risk analysis is conducted on the use cases to identify security requirements and gaps in current security and dependability technologies, standards and regulation; the project will research and develop solutions that can be used to fill these gaps, which will be tested in an in-project built security integration test environment.

SPARKS

The SPARKS project will "promote awareness of existing and emerging smart grid cyber security risks" [57]. It will develop procedural and technical countermeasures against these risks, such as cyber attack-resilient control systems, real-time network monitoring of SCADA-based control systems and novel hardware security technologies for smart metering applications. It will also provide guidance on privacy related issues, considering existing legislation.

SCISSOR

The SCISSOR project proposes "a new generation SCADA security monitoring framework", comprising four layers: i) a monitoring layer supporting traffic probes providing programmable traffic analyses up to layer 7, new ultra low cost/energy pervasive sensing technologies, system and software integrity verification, and smart camera surveillance solutions for automatic detection and object classification; ii) a control and coordination layer adaptively orchestrating remote probes/sensors, providing a uniform representation of monitoring data gathered from heterogeneous sources, and enforcing cryptographic data protection, including certificate-less identity/attribute-based encryption schemes; iii) a decision and analysis layer in the form of an innovative SIEM fed by both highly heterogeneous monitoring events as well as the native control processes' signals, and supporting advanced correlation and detection methodologies; iv) a human-machine layer devised to present in real time the system behavior to the human end user in a simple and usable manner" [58]. The proposed framework will be designed for resilience and reliability. Its results will be assessed in a real smart grid.

Assessing the Physical Impact of Cyber Threats

[59] proposes "an online framework for assessing the operational reliability impacts due to threats to the cyber infrastructure". Cyber attacks are modeled in attack trees, which represent the different paths that the attacker may follow from the first successfully exploited vulnerability to the compromise of a physical asset; e.g., from compromising the communications network to causing a power outage. A different probability and impact is associated with each path, which can be updated online with operational data. This framework helps to understand what cyber assets are more likely to be attacked and how these attacks may affect the physical infrastructure.

[60] proposes "an integrated cyber-power modeling and simulation testbed" for analyzing the impact of cyber events on the power grid. Similar to the afore mentioned work, its goal is also to help understand the relationship between power systems and cyber systems in a power grid.

Dependable Smart Grid Systems

[61] assesses a set of EDP Distribuição's smart grid systems performance and dependability requirements, compares a set of intrusion tolerant protocols to each other and to the assessed requirements and proposes a cost-benefit efficient solution for the replication and redundancy of these systems, including the SCADA, DMS and SC components.

Chapter 3

Design

This chapter proposes a knowledge-based, secure and dependable SHS architecture for smart grids. It starts with the definition of the research problem, the high level requirements, assumptions, constraints and hypotheses. Then, it breaks down the functional requirements and proposes a set of compliance providing solutions, setting the grounds for the proof of the research hypotheses. It extends the functional design is extended by presenting a smart grid model from the DSO perspective, including all smart grid technical layers. In addition to this, it presents process models for the smart grid and its dependent processes. It breaks down the security and dependability requirements by performing a threat and vulnerability assessment to the self-healing application and process, proposing a set of controls and corresponding solutions. Finally, it proposes a SHS and SHEE architecture, combining the aforementioned set of functional, security and dependability solutions.

3.1 Chapter Overview

The electrical network self-healing and security automation solutions presented in Sections 2.6 and 2.7 are restricted in the decisions they make and in the actions they take by resorting to information about a specific smart grid technical layer, disregarding related information from the other layers. In this regard, we propose a SHS that acquires and reasons with knowledge from the several smart grid layers to create more efficient, accurate and robust failure prevention and recovery plans, while minimizing their impact in the smart grid performance. The plans can be automatically executed, by sending commands to the corresponding smart grid systems and components, or they can be proposed to the smart grid operators. Intelligent supervision system components are used to provide monitoring, diagnosis, recovery and learning capabilities. Expert system components provide knowledge management and reasoning capabilities. Ontologies provide knowledge modeling capabilities. We model the systems and components in the several smart grid layers, focusing on their connections, functionalities and emphasizing the relationships between

the layers. We model the smart grid and its dependent processes as state machines that change state with fault and failure events and also with their reconfiguration in response to these events.

The self-healing pilot projects described in Section 2.6 are mainly focused on functional issues, which might make them disregard security-by-design benefits, compromising the overall security of eventual future roll-outs. In this regard, we employ a security-by-design approach to identify and propose the security and dependability requirements that provide it with the required protection, intrusion and fault tolerance capabilities, which is motivated by the SHS having access to confidential information and being capable to automatically control the smart grid. We perform a threat and vulnerability assessment to the SHS, focusing on the self-healing application as a target and as a possible threat to the smart grid, behaving erroneously or maliciously. The corresponding controls are identified and a subset comprised of those that need to be technically addressed by the SHS is proposed in the form of requirements to the security and dependability components. A MAS architecture with hybrid vertical "two-pass control" layered and cooperating agents provides distributed self-healing capabilities. In partitioning scenarios, the agents are capable of supervising the smart grid systems and components that remain within the same partition. The assignment of each agent to a self-healing domain provides role segregation within the MAS. A set of security features, including the cryptography, user management, event logging, backup and configuration modules, provide secure access, I/O, communications, storage, event logging, backup and configuration capabilities. BFT SMR, proactive and reactive recovery of replicas provide tolerance against faults and intrusions.

The proposed functional, security and dependability components are materialized in the SHEE architecture and corresponding modules. A SHEE is a replicated, intelligent, autonomous and cooperating agent of the multi-agent SHS. It is responsible for an assigned self-healing domain, which consists of a set of the smart grid systems, components and processes that are in its scope of supervision. It reasons with knowledge based on facts and rules. It monitors the self-healing domain, detecting faults and failures. It diagnoses faults to determine the cause and impact of occurred failures and to find possible causes for future failures. It creates recovery plans, consisting in an ordered sequence of fault removal and/or isolation actions, to recover from the observed failures and to prevent future failures. It reconfigures the smart grid based on the aforementioned plans. It learns from the past decisions and from the results and consequences of its actions. It communicates with other SHEEs in the MAS by using standard-based protocols and a standard-based language. It comprises a set of security and dependability features to prevent and tolerate faults and intrusions. It has a hybrid vertical "two-pass control" layered architecture, comprising the collectors, knowledge, monitoring, diagnosis, recovery, reconfiguration, controllers, learning, cooperation, user interface, SMR, user management, cryptography,

backup, configuration and logging modules, which are described with further detail in the remaining of this chapter.

3.2 Problem Definition

The European electricity sector is changing at several levels, as presented in Section 2.3, requiring the DSO to adapt its processes and technological infrastructure - the smart grid - to new challenges. As a result, the smart grid is becoming increasingly more complex with more and new types of systems and internal and external connectivities (see Section 2.4). In this new context, concerns with performance, privacy, security and dependability go further beyond electrical network faults, as explained in Section 2.5. Solutions such as those presented in Sections 2.6 and 2.7 are used to detect, locate, remove and/or isolate faults and prevent and recover from failures in specific smart grid domains, such as the electrical network and the communications. However, to our understanding, there are some development and operation issues that introduce limitations to a secure and dependable self-healing grid, which we present below.

1. The described self-healing pilot projects and research use cases employ only electrical network information for the monitoring and creation of recovery plans. Similarly, the cyber security and dependability solutions are based on a very specific set of computing or communications information to monitor, and they act by allowing or denying a given process or communication. Both are focused on very specific domains, disregarding information from other domains. However, as the Ukrainian incidents show [21], there can be cause-effect connections between failures in different smart grid layers. The knowledge and analysis of these connections could potentially be used to validate and improve the preventive and recovery actions. In this regard, we propose a SHS that is able to perform the required information acquisition and analysis.
2. The described self-healing pilot projects are mainly focused on functional issues, which might make them disregard security-by-design benefits, compromising the overall security of eventual future roll-outs. The presented European projects are assessing the security control requirements for several smart grid related use cases, in which self-healing is included. e-balance is developing its solutions in compliance with these requirements. SEGRID is also analyzing gaps in the available security technologies and will propose new solutions to fill certain gaps. Our solution introduces changes to the explained self-healing use cases. In this regard, we propose to perform an assessment of our SHS to identify and propose the necessary security and dependability controls.

3.2.1 Requirements

In response to the issues identified in Section 3.2, the proposed SHS must comply with the following high level requirements:

1. Automatically prevent and recover from failures, through fault detection, location, removal and/or isolation in all smart grid technical layers;
2. Prevent its own erroneous and malicious behavior, through security fault prevention and tolerance in its internal components;
3. Minimize its impact in the performance of other smart grid systems;
4. Operate in different user selectable modes;
5. Be flexible¹ and agile².

By complying with the requirements above, the proposed SHS should provide a more efficient, secure and dependable self-healing capability.

3.2.2 Assumptions and Constraints

The following assumptions and constraints must be considered in the design:

- In regard to the Requirement 1:
 1. The scope is the faults which are reflected in the smart grid systems and components configurations;
 2. Failures are caused by one or more faulty configurations;
 3. The current configurations can be retrieved from the systems and components;
 4. Configuration changes generate events that are sent to a collector or registered in a retrievable log;
 5. Failures generate events that are sent to the SHS;
 6. Events and retrieved configurations have two timestamps: one assigned when the event was created or the configuration was applied and a second assigned by the SHS when they are received;
 7. There is a time window within which a set of failure related events is received or retrieved by the SHS and that is sufficient to detect and locate the corresponding faults;

¹Flexibility is the "capability to adapt to new, different and changing requirements" [62], enabling the handling of predictable changes.

²Agility is the ability to thrive and prosper in an environment of constant and unpredictable change [63], enabling the handling of uncertain and uncontrollable changes.

8. The actuators are capable to apply a set of requested configurations in the order by which they are produced.
- In regard to the Requirement 2:
 1. The faults and intrusions affecting the SHS can be determined by performing a generic threat and vulnerability assessment;
 2. The risk assessment of the smart grid systems and components that are used as information sources and controllers within the SHS is handled outside of the current scope of work;
 3. The low level security and dependability requirements are the controls required to deal with the identified threats and vulnerabilities;
 4. The external controls are to be handled outside of the current scope of work;
 5. It is possible to change the information sources and controllers to support the SHS security and dependability controls, which is handled outside of the current scope of work.

3.2.3 Hypothesis Refinement

The design should support the following hypothesis:

- If the SHS is operating correctly and with the right knowledge about the smart grid systems and components, faulty configurations can be detected, located, removed and/or isolated by comparing the expected configuration with the current configuration and by comparing the expected behavior with the failure behavior.

3.3 Functional Design

The main SHS component is a software application. Its main building blocks are identified through the breakdown and analysis of the high level requirements. Requirement 2 is excluded from the following analysis, as it will be properly addressed in Section 3.6.

3.3.1 Requirements Breakdown

The high level requirements identified in Section 3.2.1 can be broke down in the following low level requirements:

1. Automatically prevent and recover from failures, by detecting, locating, removing and/or isolating faults in all smart grid technical layers;
 - (a) Collect real-time health, performance and security related events from all the smart grid technical layers;

- (b) Recognize the collected events as configuration, fault or failure related;
 - (c) Discover the cause-effect chain between events;
 - (d) Identify the best sequence³ of fault removal and/or isolation maneuvers - the recovery plan - in response to failure events;
 - (e) Consider the predictable consequences of the possible actions;
 - (f) Consider the previous success/failures of applying a given action;
 - (g) Dispatch the recovery plan in an iterative and controlled manner;
3. Minimize the impact of the SHS in the performance of other smart grid systems;
- (a) Minimize the network and the processing time utilization in other smart grid systems, by decreasing the number of information and control requests to other systems;
4. Operate in different user selectable modes;
- (a) Automatic mode, where the SHS creates and applies the recovery plan;
 - (b) Support a proposal-only mode, where the recovery plan is suggested but not applied;
 - (c) Give priority to the operators' actions, where commands issued by the operator take precedence over the actions of the proposed recovery plan;
 - (d) Support a safe shut down;
5. Be flexible and agile;
- (a) Support the deployed types of systems, components, communication protocols and events;
 - (b) Adapt to new types of systems, components, communication protocols and events;
 - (c) Adapt to unexpected event sequences.

3.3.2 Intelligent Supervision System Components

Intelligent supervision system components, namely the dispatch, monitoring, diagnosis, recovery and learning modules, give partial compliance with the Requirements 1a to 1g, which is complemented by the remaining proposals. Intelligent supervision systems can be used to timely handle different asynchronous events in real-time systems. It is comprised of four core modules, as depicted in Figure 3.1, which are described below.

³The best sequence minimizes the visibility and impact of the smart grid reconfigurations to the customers and users and to the network, respectively.

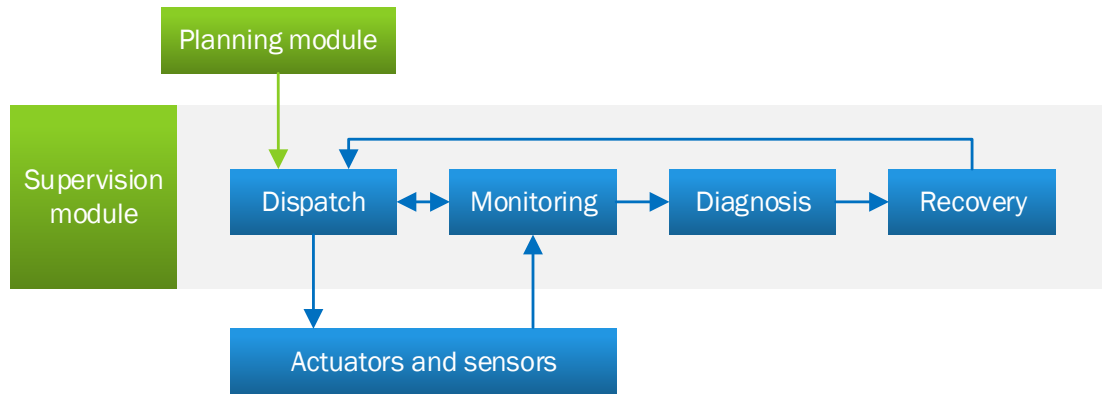


Figure 3.1: Information flow in a supervision architecture.

- Dispatch - Executes a plan given by the planning module, by sending commands to the device controllers;
- Monitoring - Monitors a process, based on the information retrieved from sensors, in search of deviations from the plan objectives, problems or abnormalities;
- Diagnosis - Analysis the identified issues and identifies their cause;
- Recovery - Tries to recover from the issue by generating a recovery plan.

An intelligent supervision system may further include the following functionalities.

- Prognosis - Analyses the temporal evolution of a set of state variables and tries to anticipate faults and consequent failures;
- Preventive maintenance - Generates preventive maintenance plans;
- Learning - Improves the performance of the previous functionalities through the acquisition of knowledge to be used in future events.

3.3.3 Expert System Components

Expert system components, namely a knowledge base and an inference engine, add compliance with the Requirements 1b to 1e and 5c. An expert system is "a computer program that behaves like a human expert in some useful ways" [64]. Artificial intelligence techniques are used to create an expert system, such as the modeling of knowledge through a set of logical propositions and the capability to reason with that knowledge through the use of rules. The core architecture is comprised of:

- Knowledge base - Contains the modeled knowledge;
- Inference engine - Reasons with the modeled knowledge;

These are complemented by:

- User interface - Interacts with the user;
- Data base - Stores relevant data.

A real-time expert system is an expert system that has the capability to deal with real-time information.

3.3.4 Knowledge Modeling

An extensive and deep understanding of the supervised systems, components and processes and the selection of an extensible knowledge representation adds compliance with the Requirements 5a and 5b. In this regard, Sections 3.4 and 3.5 present simplified models of the different smart grid layers and processes, which can be used as a base for the specification of a smart grid self-healing ontology. An ontology is a formal specification of a set of terms used to describe and represent an area of knowledge, being widely used in knowledge engineering, artificial intelligence and computer applications that involve knowledge management and information management [65].

Ontologies have been used in smart grid security related work. For example, in the SPARKS project, ontologies are used twofold. In Deliverable 2.2, an ontology is used to represent semantic threat graphs, describing the relationships between incident scenarios and security requirements and controls. Through reasoning, it is possible to select the most adequate countermeasures for each specific incident scenario [66]. Deliverable 2.3 proposes an ontology for describing the Smart Grid Architecture Model (SGAM)⁴. Ontology reasoning is used to automatically identify threats, analyze vulnerabilities, support likelihood assessment and, as a result, automatically generate attack trees [68].

Two main ontology language specifications were developed by the World Wide Web Consortium (W3C): Resource Description Framework (RDF) and Web Ontology Language (OWL).

- RDF - It is a standard for data interchange in which information is encoded through the construction of triple structures with resources, properties and literals/values, identified by Uniform Resource Identifier (URIs), resulting in a directed labeled graph where the nodes are resources and the edges are the properties [69].
- OWL - It is used in information processing applications, having the capability to describe classes, properties and values of concepts and the relationship between them. Restrictions can be defined for classes. Characteristics and constraints can be

⁴The SGAM is part of the Smart Grid Reference Architecture presented by the Smart Grid Coordination Group (SGCG). It describes the smart grid as a three dimensional infrastructure, consisting of different domains, zones and layers [67].

defined for properties. A knowledge base of a given knowledge domain is populated with individuals created through the instantiation of the classes and properties of the corresponding ontology. Classes, properties, values and individuals are identified by the corresponding URIs [70].

3.3.5 Information Collection and Control

The possibility to add and remove information collectors and controllers from the SHS gives compliance to the Requirements 1a and 1g. The collectors and controllers must implement the different communication languages and protocols used by the various smart grid systems and components. The use of collectors is a common approach, for example, in the case of the SCADA system where the frontends obtain and consolidate data gathered from distinct parts of the infrastructure, and the SIEM system with the connectors. In both cases, the collectors are used to gather information from the sources and translate it to their internal models.

In the smart grid environment and as explained in Sections 2.4 and 2.7, the information is commonly concentrated in several servers for centralized processing. This happens, for example, in the RTUs, SCs, SCADA servers and SIEM. Regarding the control capability, although the RTUs and IEDs can be directly controlled, the most secure and safe way to proceed is to use the SCs for local control and the DMS and SCADAs servers for remote control. Giving preference to these data sources and controllers for information collection and control adds compliance with the Requirement 3a.

3.3.6 User Interaction

A user interface for local and remote management, operation and maintenance support, with the possibility of selecting several optional supervision modes, adds compliance with the Requirements 4a to 4d. The options to set the SHS to proposal-only mode or to give priority to user commands, passing over its own, are relevant for building the operators' trust in an early operation stage. The option to partially or completely switch off an erroneous or malicious SHS is essential for the safety of the smart grid.

3.4 Smart Grid Model

The smart grid is comprised of different infrastructural layers, which connect the electrical equipment to the control systems, to enable the automation and the remote control of the electrical network. Figures 3.2 to 3.4 present in more detail a view of the connections among the different layers.

3.4.1 Electrical Network

A simplified model of the electrical system is depicted in the Figure 3.2, focusing on the electrical connections and electrical switching capabilities of the electrical network, observed from the perspective of the DSO. In this regard, the electricity distribution network is depicted with a higher detail than that of the remaining components, here represented for two primary substations, three secondary substations and four DA devices, and its connection with the transmission network. The figure contains representations of power transformers, lines, busbars and switchgear.

- Transformer - The purpose of a transformer is to convert a higher voltage to a lower voltage (i.e., HV to MV or MV to LV) or vice-versa, depending on the current flow. Some transformers have a tap changing mechanism, which enables the regulation of the output voltage through the selection of different winding taps. The taps are connection points located along a transformer's winding that account for a certain number of turns and, consequently, different conversion ratios. The transformers are labeled in the figure with "TRx".
- Switchgear - An electrical switch is a mechanical switching device that enables the electrical isolation and/or protection and/or operation of the electrical network and loads, such as motors, heaters, lighting and capacitors. A switch has typically two configurations: opened and closed, meaning disconnected or connected, respectively. Both configurations require the internal mechanism to be in a specific physical position. The switch is malfunctioning when the mechanism is not being actuated and it is not in one of the two aforementioned positions. The electrical switches are labeled in the figure with "ESx".
- Line - A line allows the flow of current between the loads that are electrically connected in both of its ends. They are represented by the mainly horizontal lines that connect the other representations. The dashed lines mean that in the real world there would be other components in the middle, which were not drawn due to their similarity to the represented components. The lines are not labeled in the figure.
- Busbar - A busbar allows the flow of current between the lines that are connected to it. They are represented by the vertical lines that connect sets of lines. The busbars are not labeled in the figure.

3.4.2 Communications Network

A simplified communications network model is depicted in Figure 3.3, further narrowing the focus to the connections between substations and DA. The figure contains representations of switches, routers, firewalls and physical connections.

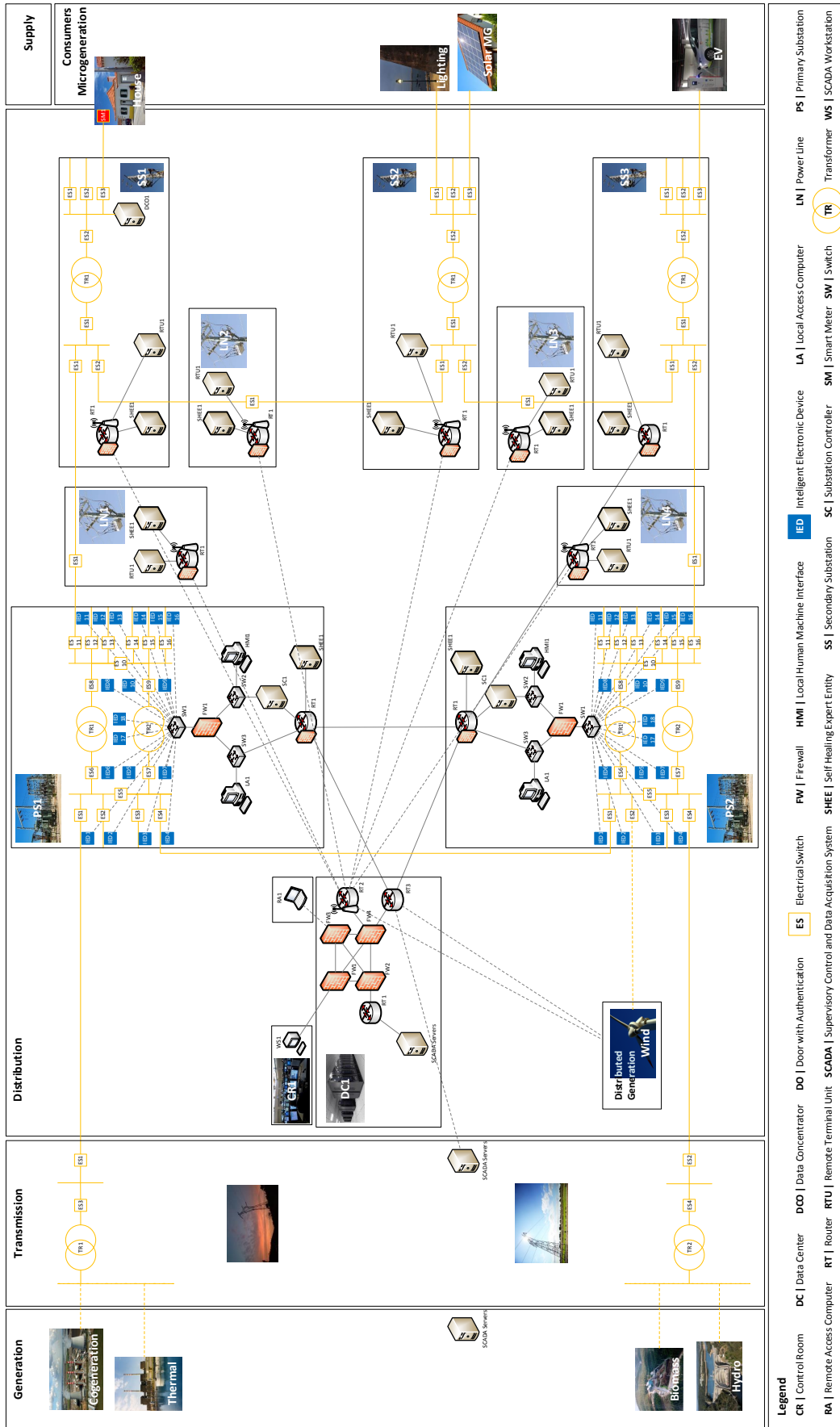


Figure 3.3: Communications network model.

- Switch - A switch is used to expand a network, enabling communications between the network devices that are connected to its network interfaces. They may support the configuration of VLANs, which allow network segregation through the separation of the physical network interfaces by different logical networks. Switches are labeled in the figure with "SWx".
- Router - A router is used to create a network and/or to enable the communication between the networks that are connected to its network interfaces. They support the configuration of static IP routes and dynamic routing, which allow them to decide which IP communications should be sent through which IP network interfaces. Routers are labeled in the figure with "RTx".
- Firewall - A firewall is used to allow/prevent the network devices from sending/receiving any or a set of IP communications to/from other network devices through its network interfaces. They support the configuration of rules for each interface, which can be based on the IP, protocol and content of the communications. Firewalls are labeled in the figure with "FWx".
- Physical connections - Regarding the physical connections, the figure contains representations of landlines, wireless and PLC. The landlines cross the boundaries of the facilities, connecting them to each other. The dashed lines represent the wireless connections. The PLC is represented by the electrical network components that connect the secondary substation SS1 to the house in the upper-right corner of the figure. It works in each electrical phase, between the transformer and the smart meter. Inside each facility, the connections between the network devices are represented by simple lines. The physical connections are not labeled in the figure.

3.4.3 SCADA and Automation

The SCADA and automation components and a simplified service model are depicted in Figure 3.4, with the connections between the components drawn in blue. The SCADA system collects, processes, stores and displays the data from the electrical network. It also enables the operators to consult and analyze the data and to give commands to the electrical network. Automation devices have the capability to automatically process the data, to coordinate between themselves and to autonomously execute predetermined actions on the electrical network.

- IED and RTU - The intelligence of the smart grid begins at the IEDs in the primary substations and the RTUs in the secondary substations and MV distribution lines. They collect the data directly from its source, they process the data and they execute a predetermined action, which may simply be the sending of the data and/or any additional information that was generated from the data processing to the central

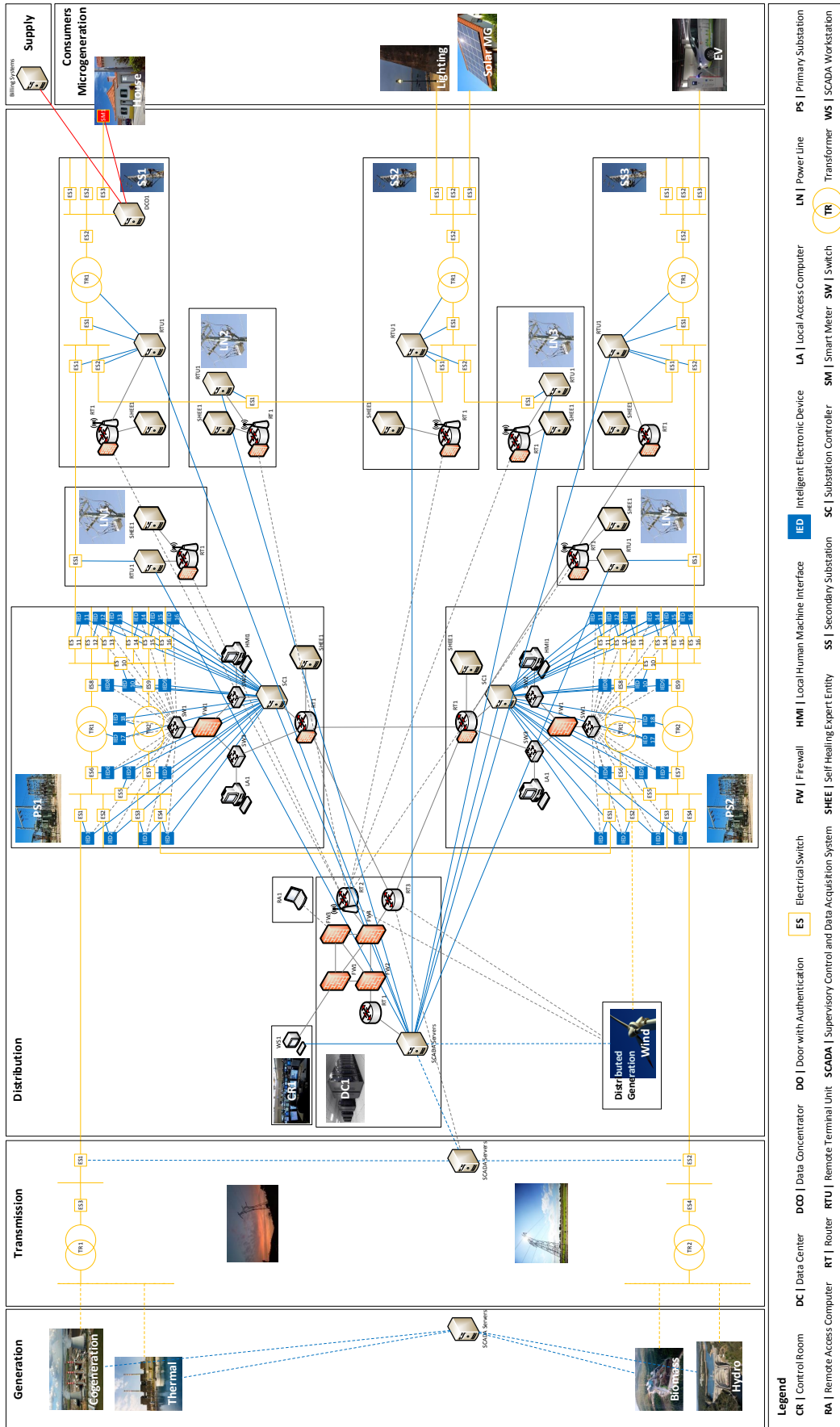


Figure 3.4: SCADA and automation.

systems, or it may be the sending of a command to an actuator. They provide also an integrated user interface for manual control of the electrical network. If they are configured to local mode, they accept only commands given through this interface. If they are configured to remote mode, they accept commands given by the SC. IEDs and RTUs are labeled in the figure with "IEDx" and "RTUx", respectively.

- SC and HMI - In a primary substation, the data sent by the IEDs is collected by the SC, providing a local view of the electrical network within the facility. This data is sent by the SC to the SCADA servers but it can also be monitored locally through the HMI. It is also possible to set the substation to local or remote operation modes through the HMI interface. The SCs and the HMIs are labeled in the figure with "SCx" and "HMIx", respectively.
- SCADA (and DMS) servers - At the data center, the SCADA servers receive the telemetry data from the substations and from the DA devices, which they process, store and make available to the operation workstations for consultation. They also receive, validate, translate and forward the operators' commands from the workstations to the substations and DA devices.
- SCADA workstations - In the control room, the operators have access to the workstations, which connect with the SCADA servers to display the smart grid information and to send commands to the substations and DA devices. The workstations are labeled in the figure with "WSx".

In regard to security, we consider that the aforementioned components support addition and removal of users and roles, whitelisting, firewall rules and antivirus signatures.

3.4.4 Further Considerations

For design purposes, it is considered that the following systems and components are also present in the smart grid, although some are not represented in the aforementioned Figures:

- Electricity supply to electronic equipment (inside the DSO's facilities) - The different facilities provide, with different redundancy levels, the LV electricity supply required by the hosted systems and components. For instance, PS1 might have MV to LV auxiliary transformers and it might also have electrical connections to other substations, together with the corresponding SCADA components. There are also local and central systems that allow to monitor and control of these electrical network components.
- Security and dependability controls - There are other security and dependability controls are in place, supporting the configuration of rules, signatures, policies and

use cases, in addition to those that were already mentioned. For example, network segmentation and segregation, hardening and intrusion detection and prevention systems and redundant components and communication channels.

- Sensors - There are several sensors spread through the smart grid infrastructure.
- Supported services - The smart grid provides other services beyond SCADA, such as: local and remote engineering (through the engineering computers LA1 and RA1, respectively), energy quality control (not represented) and smart metering (through the data concentrator DCO1 and smart meter SM1).

3.5 Processes Model

The DSO manages a set of processes that support and enable its role in the electricity distribution activity and which depend on the smart grid systems and components to execute correctly. To further clarify the existing dependencies, the smart grid and a generic smart grid dependent process were modeled as state machines, which are depicted in the Figures 3.5 and 3.6.

3.5.1 Smart Grid

By increasing the abstraction level on the work presented in Section 3.4, it is possible to represent the smart grid as a sequence of states and transitions, as depicted in the Figure 3.5. The smart grid state at any given moment is the combination of the states of its components. The state of a smart grid component is determined by comparing its health (i.e., operational or not operational), configuration and performance (e.g., current, voltage, temperature, used memory, used storage and used processing resources) with the planned configuration and with the rated performance values (e.g., rated current, rated voltage, rated temperature, memory capacity, storage capacity and processing capacity). Regarding the planned configuration and using the electrical network as a first example, the command center (in the control room) determines which is the optimal network configuration for a certain period, given a set of conditions (e.g., the expected transmission delivery, the expected distributed generation, the expected MG, the expected distribution supply, the programmed network interventions, the critical consumers and the regulation). As a second example, the routers, switches and firewalls are installed and setup with a set of routes (or a dynamic routing protocol), VLANs and rules, respectively, in compliance to their use case and to the organization's policies.

- Planned configuration - When all the components are in a planned configuration and there are no programmed or fault-imposed restrictions to the performance of their functions;

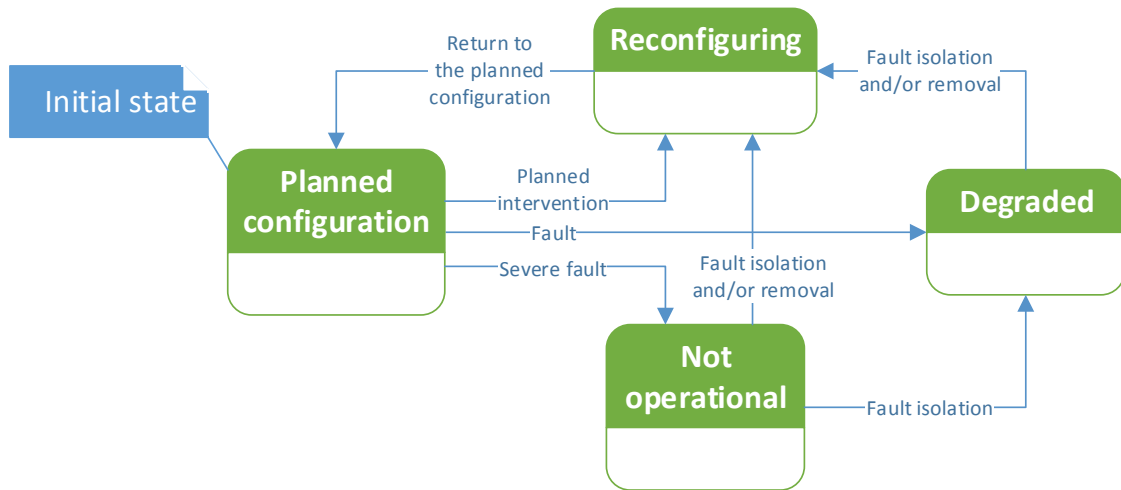


Figure 3.5: Smart grid operation model.

- If there is a planned intervention in which at least one component is reconfigured to an alternative configuration, then the smart grid will change to the "reconfiguring" state;
 - If a fault affects at least one component in a way that it is still able to perform its basic functions, then the smart grid will change to the "degraded" state;
 - If a fault affects a set of components in a way that they are not able to perform their basic functions, leaving a part or the total smart grid inoperative, then the smart grid will change to the "not operational" state.
- Reconfiguring - When the smart grid is operating in an alternative configuration because it is being intervened (e.g., a set of clients that are being supplied through an alternative substation, or the routing of the communications through an alternate path);
 - When the intervention ends, the smart grid will return to the "planned configuration" state.
 - Degraded - When fault-imposed restrictions prevent the smart grid from returning to the planned configuration (e.g., an overheating electrical line, a "slow" SCADA server or an isolated fault);
 - The smart grid will change to the "reconfiguring" state when the fault is isolated and/or removed, which may require several steps within this state.
 - Not operational - The smart grid is not operational and in no condition to support the dependent processes (e.g., a broken electrical line, or a damaged firewall);

- The smart grid will change to the "reconfiguring" state when the fault is correctly isolated and/or removed, which may require several steps within this state;
- As the smart grid is unable to support any process execution, it is a priority to put it into operation. Therefore, the smart grid will change to the "degraded" state right after the fault isolation, if possible.

3.5.2 Smart Grid Dependent Processes

A DSO managed process depends on the smart grid systems and components if there are smart grid states and transitions that have a disruptive/healing impact in its execution. For instance, the electricity distribution core business process depends on the state of the electrical network layer and on the state of the electrical network control process. The electrical network control process depends on the state of the communications network layer and on the state of the SCADA systems, components and service. Figure 3.6 depicts the states and transitions of a generic smart grid dependent process in relation to the smart grid states. The application of this model to the analysis of a specific process should consider only the smart grid systems and components that support the analyzed process's execution.

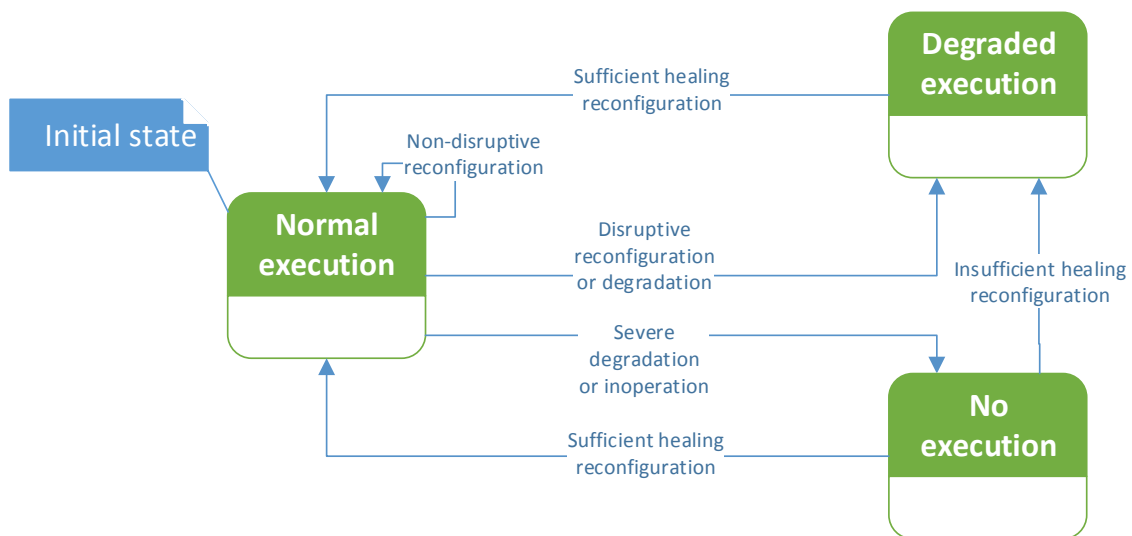


Figure 3.6: Generic smart grid dependent process model.

- Normal execution - When the process is executing without restrictions;
 - If the smart grid changes to the "reconfiguring" state but the reconfiguration is non-disruptive to the process, then the latter will remain in the "normal execution" state;

- If the aforementioned reconfiguration is disruptive to the process, or if the smart grid changes to the "degraded" state but it can still support the process with limitations, then the process changes to the "degraded execution" state;
 - If the smart grid changes to the "degraded state" and it is unable to support the process, or if the smart grid changes to the "not operational" state, than the process changes to the "no execution" state.
- Degraded execution - When the process is executing with restrictions (e.g., when a primary substation cannot be remotely controlled and, consequently, the local HMI must be used);
 - If the smart grid is reconfigured in such way that it can support the process without restrictions, the latter changes to the "normal execution" state.
 - No execution - When the process is not executing;
 - If the smart grid is reconfigured in such way that it can support the process without restrictions, the latter changes to the "normal execution" state;
 - If the smart grid is reconfigured in such way that it can support the process with restrictions, the latter changes to the "degraded execution" state.

3.6 Security and Dependability Design

The security and dependability building blocks are identified through the breakdown and analysis of the high level Requirement 2 - Prevent the erroneous and malicious behavior of the SHS, through security fault prevention and tolerance in its internal components. In this regard, a generic threat and vulnerability assessment of the Self-Healing Application (SHA) was performed to identify the required security and dependability controls and to propose a set of compliant and compliance gaining components. In regard to the proposed components, these focus only on the controls that have to be implemented through SHA components or to which it must provide support. Other external controls are not covered by this investigation.

3.6.1 Threat and Vulnerability Assessment

The assessment starts with the identification of the relevant SHS assets, to which follows the identification of the relevant threats and vulnerabilities. It is assumed that the existing smart grid assets have already been subject to a similar assessment. Therefore, we focus our efforts on the threats to the SHA and also on the threat that a breached, erroneous, leaking or malicious SHS might mean to the smart grid.

Assets

Table 3.1 contains a list of the SHS relevant assets, including a small description of the reason why they are considered important. The list includes several types of assets, including the SHA, people, hardware, third-party software, information, services and facilities.

Table 3.1: Assets table.

ID	Asset	Description
01	Facility	The facility where the self-healing application is hosted
02	Hardware	The hardware that supports the self-healing application (e.g., rack and computer hardware)
03	Hypervisor	The hypervisor that supports the self-healing application
04	OS	The OS that supports the self-healing application
05	Application	The self-healing application
06	Information	The information that is handled by the self-healing application
07	Communications	The communications between the application and the data sources/controllers
08	Data Sources and Control Devices	The smart grid sensors, data concentrators, actuators and control systems
09	Users	The self-healing management and maintenance teams
10	Processes	The self-healing management, operation and maintenance processes

Threats and Vulnerabilities

Table A.1, in Appendix A, contains an extensive list of the threats and corresponding vulnerabilities that might target the SHA and the smart grid, through it. The aforementioned list was compiled by combining our expertise on the matter with available lists of common threats and vulnerabilities, such as [71]. It is not final due to the fact that new threats may arise and new vulnerabilities may be discovered that target the smart grid and the SHA in specific. Also, different SHS deployments in different smart grid contexts might expose the SHA to specific threats unknown to us at the current time.

The threat-vulnerability pairs can be divided in two main groups, regarding their consequences to the SHA and to the smart grid. Line IDs with letters A and B present the threats and vulnerabilities with immediate consequences to the SHA. If the corresponding incident is not contained, the smart grid might also be affected. Line IDs with letters C and D present the threats and vulnerabilities that immediately affect the SHA and the smart grid. They are mainly related with the erroneous or malicious behavior of the SHA and the compromising of the handled smart grid information.

3.6.2 Controls

In Table A.1, in Appendix A, we propose controls for all the identified threat-vulnerability pairs, without exclusions. These controls were compiled by combining our expertise on the matter with available lists of security and dependability controls, such as the ISO/IEC 27002:2013 standard⁵. The DSOs or SHS manufacturers are encouraged to complete the risk assessment, beyond the threat and vulnerability assessment, to select which of the identified controls are applicable to their smart grid context. If necessary, the proposed controls might be replaced by equivalents. The controls can be divided in two main groups, separating the SHA internal controls from the external controls.

Internal to the SHS

Line IDs with letters A and C present the controls that have to be implemented through SHA components or which it must support. They include the following:

1. Distributed application architecture;
2. Access:
 - Local user accounts;
 - Centralized personal user accounts with fallback to the local user accounts;
 - Fine-grained RBAC;
 - Enforce compliance with strong password policy requirements, such as: reuse of old passwords, maximum and minimum age, minimum length and complexity;
 - Automatic idle user logout;
3. I/O:
 - Automatic input and workflow restriction, validation and confirmation;
 - Regular proactive data collection and update;
 - Redundant data sources and control devices;
 - Data comparison between redundant data sources;
 - Command success confirmation;
4. Communications:
 - Error detection and correction mechanisms;

⁵The ISO/IEC 27002:2013 standard contains a list of mandatory controls for an ISMS implementation, as defined in the ISO/IEC 27001:2013 [72, 73].

- Secure hashing algorithms and hash comparison;
 - Secure authentication protocols;
 - Secure key exchange protocols;
 - Application level communications encryption;
 - Secure message flow validation;
5. Storage:
 - Application or OS level storage encryption and integrity protection;
 - Cryptographic key vault;
 6. Security event logging and monitoring;
 7. Automatic and on-request backups and backup storage protection;
 8. Secure remote configurations and updates;
 9. Secure coding practices;
 10. Active replication in different VMs, physical hosts or physical geographies;
 11. Proactive and reactive system recovery.

External to the SHS

Line IDs with letters B and D present the controls that have to be implemented externally to the SHA, which does not decrease their relevance. They include the following:

1. People:
 - Regular HR capacity review;
 - HR recruitment and selection criteria;
 - User training;
 - Security awareness training;
2. Processes:
 - Regular user management policy review;
 - Regular user account and permission review;
 - Regular capacity management policy review;
 - Regular change management policy review;
 - Equipment disposal procedures;

- Regular configuration procedures review;
- Regular update procedures review;
- Regular incident response policy review;
- Regular business continuity plan review;
- Regular information classification policy review;

3. Technology:

- Physical security logging and monitoring;
- Health and performance monitoring systems;
- Whitelisting, antivirus, host and network-based firewall, intrusion detection and prevention;
- Physical , VPN, VLAN and VRF configurations;
- Data Loss Prevention (DLP) systems;
- Redundant power and communications supply;
- Redundant secure communications channels;
- Data store redundancy;

4. Services:

- Robustness testing and evaluation;
- Regular penetration testing and vulnerability management;
- Health and performance monitoring;
- Manufacturer and/or vendor contractual support;

3.6.3 Distributed Application

A MASs architecture provides compliance with Control 1, while being compliant with High Level Requirements 1 and 5. In [74], an agent is defined as "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives". Its behavior depends on how it perceives the environment, based on an internal representation, on the decisions it is prepared to make and on the actions it is programmed to take. In this regard, an agent can be classified as deliberative, reactive or hybrid.

- Deliberative - An agent that behaves pro-actively towards the achievement of a pre-determined goal. It maintains a representation of the world and, through reasoning, it is capable of planning a course of action, which comprehends the generation of

a correct and optimal sequence of actions that take him closer to its goal. It can be achieved through a Belief-Desire-Intention (BDI) architecture in which the reasoning process takes into consideration the following representations:

- Beliefs - Knowledge about the agent itself and its environment;
 - Desires - Goals that the agent has to achieve, but it still does not know how;
 - Intentions - Goals with which the agent is committed and that it knows how to achieve.
- Reactive - An agent that responds with robust actions to changes in the environment, as they happen. Unlike the deliberative agent, it does not have an internal knowledge representation nor is it capable of anticipation. It assumes that intelligence is in the world - not inside the agent - and that intelligent behavior "emerges" from the interaction between both. It can be achieved through a hierarchy of behaviors architecture in which each behavior has a situation-action rule-like structure. In this architecture, behaviors compete with each other, with lower-layer behaviors, which represent more "primitive" kinds of behavior, having precedence over higher-layer behaviors.
 - Hybrid - An agent that combines the characteristics of deliberative agents and reactive agents. It can be achieved through a multi-layered architecture in which some layers are deliberative and the others are reactive, being capable to provide optimal action sequences and fast responses. It can assume two main configurations:
 - Horizontal layering - The sensors and actuators are directly connected to the different layers, which can result in different commands being given to the same actuator, at the same time.
 - Vertical layering - The sensors and actuators are connected in two main configurations. In "one pass control", the sensors are connected to the bottom layers, the actuators are connected to the top layer and the information flows bottom-up. In "two pass control", the sensors and actuators are connected to the bottom layer and the information flows up and then down through the layers. In both, only one layer controls the actuators, preventing concurrency between the layers. However, one failed layer stops the whole process.

In a MAS, the agents interact with each other, requiring the capability to:

- Coordinate - To access non-sharable resources;
- Cooperate - To achieve a common goal by working together;
- Negotiate - To reach agreements on common interest matters.

These capabilities are also essential for the participation of the agents in coalitions. A coalition is a structure that results from the association of at least two cooperating agents. Negotiation between the agents is required for autonomously forming the coalition. Coordination of one-another actions is required for performing a joint activity. The agents communicate through Agent Communication Languages (ACLs) and agent communication protocols.

- ACLs - They are used to provide message exchange formats. The main ACLs are Knowledge Query Manipulation Language (KQML), developed in the ARPA knowledge sharing initiative, and Foundation for Intelligent Physical Agents (FIPA)-ACL, developed by FIPA. They have a similar message structure, containing a performative⁶, a content and a set of control parameters [75]. KQML includes the Knowledge Interchange Format (KIF) language to express message contents.
- Agent communication protocols - They define an ordered sequence of messages that the agents exchange when communicating with one-another. In this regard, FIPA has defined a set of protocols, including the request, contract net and subscribe interaction protocols [76, 77, 78]. The message performatives are assigned depending on the used protocol and its execution state.

In the light of the above, we propose to populate the self-healing MAS with hybrid behavior, vertical "two-pass control" layered and cooperating agents. The hybrid behavior enables a self-healing agent to follow its supervision goals and also react to smart grid events and requests from the other agents. The vertical "two-pass control" layering enables a feedback loop in the flow of information that is closed internally, by the previously presented learning functionality, and externally, through the interaction with the smart grid systems and components.

3.6.4 Self-healing Domains

The definition of self-healing domains and their assignment to different agents provides role segregation within the MAS. A self-healing domain is a set of systems, components and services of the same or different smart grid technical layers, which is assigned to an agent when it starts executing and which defines the scope for its knowledge and supervision activities. Examples of potential self-healing domains are: a component, a system, the components within a facility, an electrical network section, a set of network equipment, a microgrid or the smart grid as a whole.

⁶Examples of FIPA-ACL performatives include: request, agree, refuse, inform, failure, Call for Proposals (CFP), propose, accept-proposal, reject-proposal and subscribe.

3.6.5 Security Features

The SHA components must implement or support the controls listed in Control Groups 2 to 8 of Section 3.6.2, providing secure access, I/O, communications, storage, event logging, backup and configuration. Additionally, we propose a set of components to provide the necessary functions and automatic behaviors, namely, cryptography, user management, event logging, backup and configuration modules.

To provide compliance with Control 9, we will refer to secure coding guides, such as the OWASP Secure Coding Practice Quick Reference Guide [79], for any developed code during the implementation. Any new components or other future additions to the SHA must follow a similar approach.

3.6.6 Dependability Techniques

The replication of each agent, through the use of a BFT SMR library, provides compliance with the Control 10. It provides tolerance against faults and intrusions, which are listed as threats to the SHA. The replicas can be placed in different VMs, physical hosts or physical geographies, which must be defined by the DSO or manufacturer in each specific smart grid context. This decision must consider the physical location of the proposed self-healing domain assets, the reliability of the communication channels to the previous and the existing common threats between the self-healing domain assets and the SHA.

The redundancy and SMR of smart grid systems has been addressed in several works. However, we focus our attention on [61], which handles this issue in the context of a real deployment. It assesses the performance and dependability of a set of EDP Distribuição's smart grid systems. It compares a set of intrusion tolerant protocols to each other and to the assessed requirements. It proposes a cost-benefit efficient solution for the redundancy and SMR of these systems, including the SCADA system, DMS and primary substation RTUs. It assumes that the workstations and RTU, acting as clients, are modified to provide the necessary support.

The SHS and the SCADA system have similarities that allow us to propose BFT SMR for the agents. The SHS, like the SCADA system, has the capability to monitor and control the electrical network. It can also perform these actions automatically, while the SCADA system requires several human operators. The more advanced self-healing approaches are still in active development, being used for simpler scopes (i.e., FLIR in MV feeders). They are expected to increase in scope and complexity, as shown in the European projects' use cases, getting the operators to rely more on them, as it happened with the SCADA systems in the past. Therefore, in the future, they might match or even surpass the SCADA systems in criticality.

In regard to the above, each agent must be comprised of $3f + 1$ replicas to provide tolerance against f faults or intrusions, where f is the output of a cost-benefit analysis

to be performed by the DSO or manufacturer in their specific smart grid context. The number of replicas can be higher to account for when a replica is being recovered in compliance with the Control 11. In this regard, we assume that the rejuvenation of the agent, including the cryptographic material, is performed together with the OS through an external process. The monitoring of the replicas, through the analysis of their logs, is essential to detect faulty behavior and initiate the recovery process. Nevertheless, replicas should be regularly rejuvenated to account for false negatives in the monitoring process.

BFT SMR requires all the agents to start in the same state and to execute the same sequence of input commands, in the same order, to follow the same sequence of states/outputs, with determinism being mandatory. To comply with these requirements, all the replicas start with the same knowledge, which is comprised of an ordered set of facts and rules about the corresponding self-healing domain. It is an input from the user, which must be given to all replicas. The BFT SMR protocol total orders the information from the sensors, handling also any checkpoints and state transfers as required. The reasoning is performed deterministically through the processing of the rules in the given order, producing the same sequence of commands. Learning must also be achieved through deterministic algorithms. Therefore, approaches that rely on randomness, such as genetic algorithms, require the use of the same seed for each cycle in all replicas.

3.7 A Self-healing Expert Entity

A SHEE is a replicated, intelligent, autonomous and cooperating agent of the SHS that reasons with knowledge based on facts and rules. It is capable to monitor the smart grid systems and components, to diagnose and recover from failures detected in the latter and to learn from its actions and past decisions, within its given self-healing domain. It communicates with other SHEEs in the MAS by using standard-based protocols and a standard-based language. It comprises a set of security and dependability features to prevent and tolerate faults and intrusions. Below, we present in more detail the SHEE, including in its architecture the components proposed in the previous sections.

3.7.1 Architecture

We provide three perspectives of the architecture, starting with how the SHEEs connect with the supervised smart grid data sources and controllers to form the SHS, following to the internal architecture of a SHEE and finalizing with how the self-healing process works and how it relates with the smart grid processes.

Self-healing System

The SHEEs and their connections with the smart grid systems and components are depicted in Figure 3.7 with the label "SHEEx" and with a green color, respectively. The connections between SHEEs are represented with the same color. This representation assumes a previous decision of placing one SHEE per substation and feeder automation device. It is possible to observe differences between the SCADA and the SHS by comparing the Figures 3.4 and 3.7. The main difference is the connectivity that occurs between devices at the same hierarchical level in the SHS, which is not observed in the SCADA.

SHEE Architecture

The SHEE has a modular architecture, with the several modules corresponding to the components proposed during the previous design stages. Two separate views are provided to facilitate understanding of the proposed connections. In this regard, Figure 3.8a and Figure 3.8b depict the functional perspective and the non-functional perspective, respectively. Each box corresponds to a module, which might be an agent behavior or simply a software library that provides functions to the other modules. The arrows represent the connections between the modules, with the arrow direction indicating the main flow of information. The dashed lines correspond to configurable communications. The modules and connections are further described in Sections 3.7.2 and 3.7.3.

Self-healing Process

A simplified model of the self-healing process is presented in Figure 3.9, mapping the Fault Detection, Location, Isolation and Restoration (FDLIR) activities to the SHEE supervision activities and corresponding modules. In this model, self-healing is an ongoing process, with a monitoring activity attempting to detect eventual faults and failures within a given self-healing domain and triggering the following activities.

- Monitoring - Collects information from the smart grid sensors and analyses it to detect eventual faults and failures.
 - What is happening in the smart grid? - If a fault or failure is detected, then the smart grid is either in a "degraded" or "not operational state", depending on the severity of the fault. Isolated faults can also be detected in the "reconfiguring" state.
 - What is happening with the smart grid dependent processes? - The processes might be in "normal execution", "no execution" or "degraded execution", depending on the impact of the failure.
- Diagnosis - Performs fault location to determine the cause and impact of occurred failures and to find possible causes for future failures.

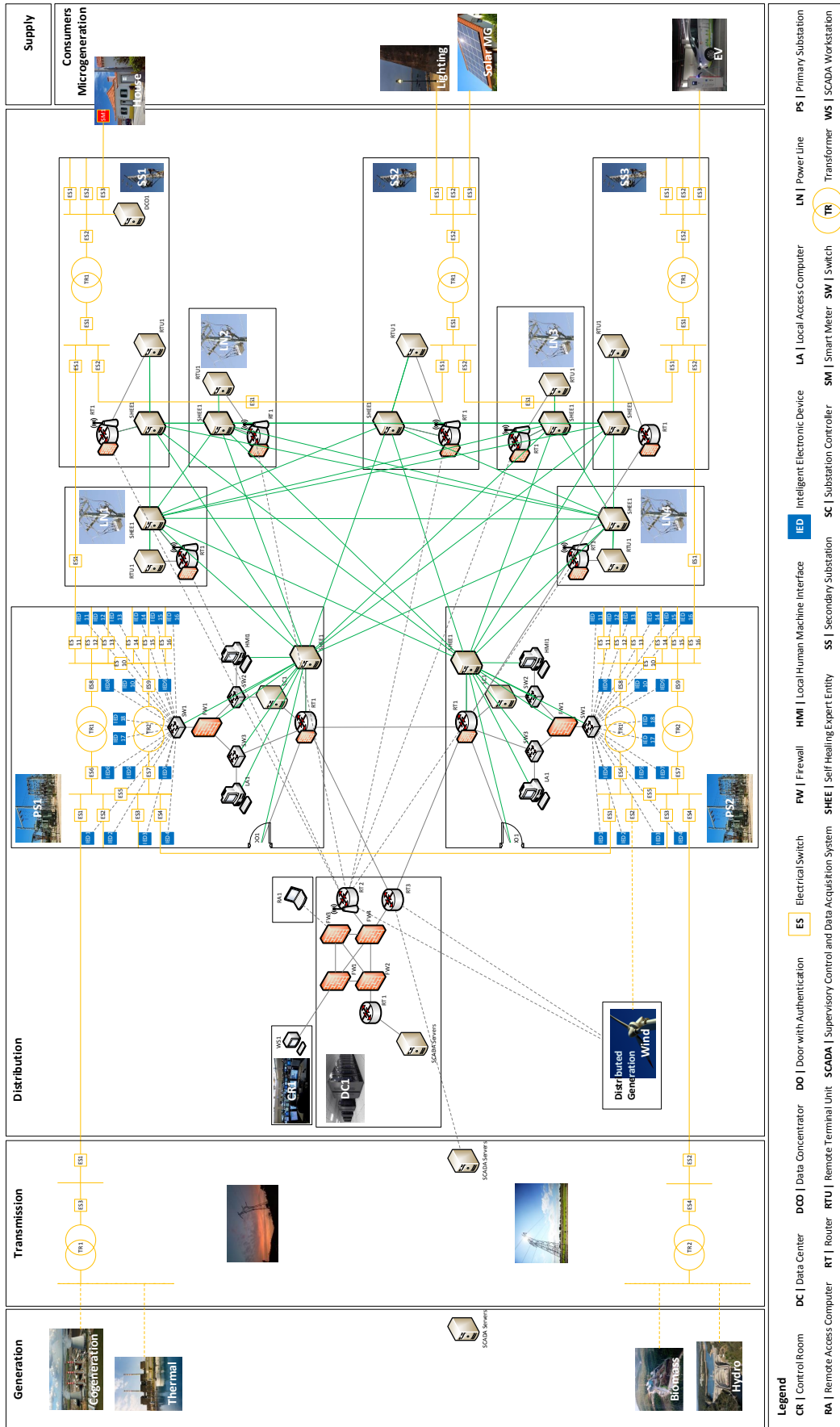
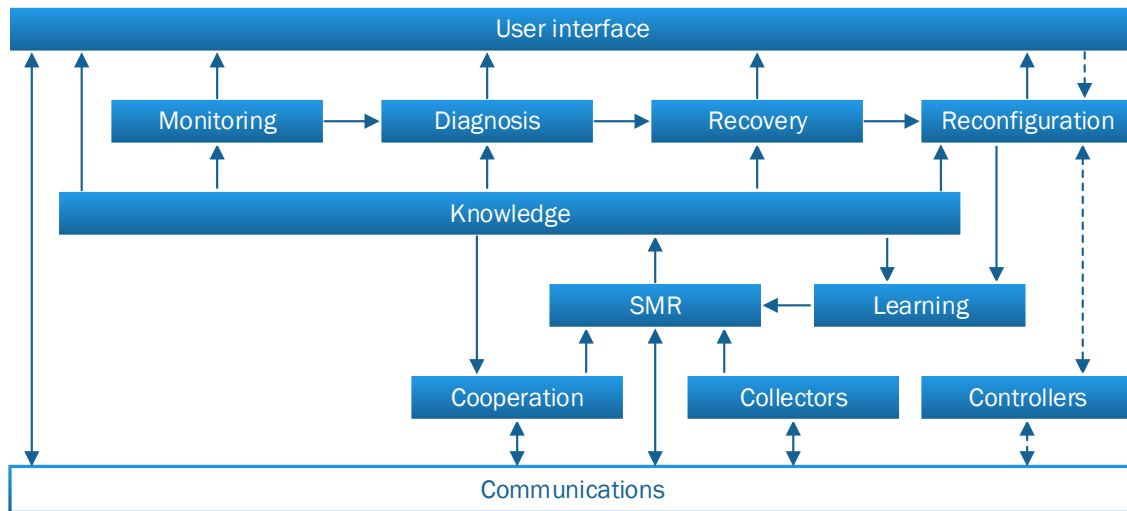
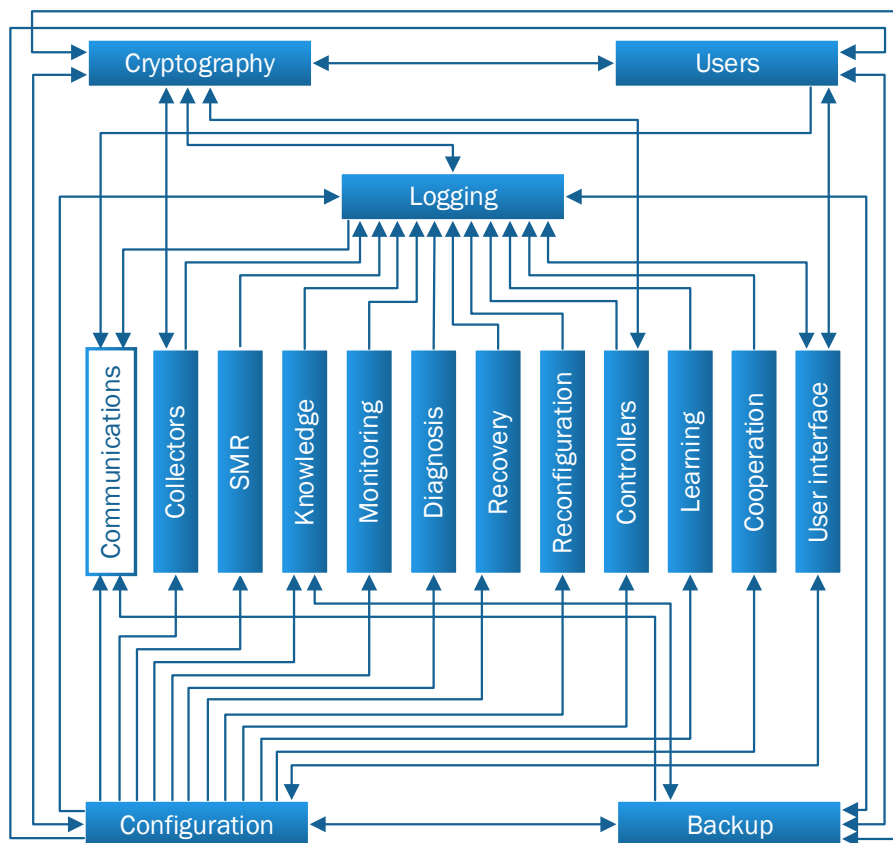


Figure 3.7: Self-healing multi-agent system.



(a) Functional perspective.



(b) Non-functional perspective.

Figure 3.8: SHEE Architecture.

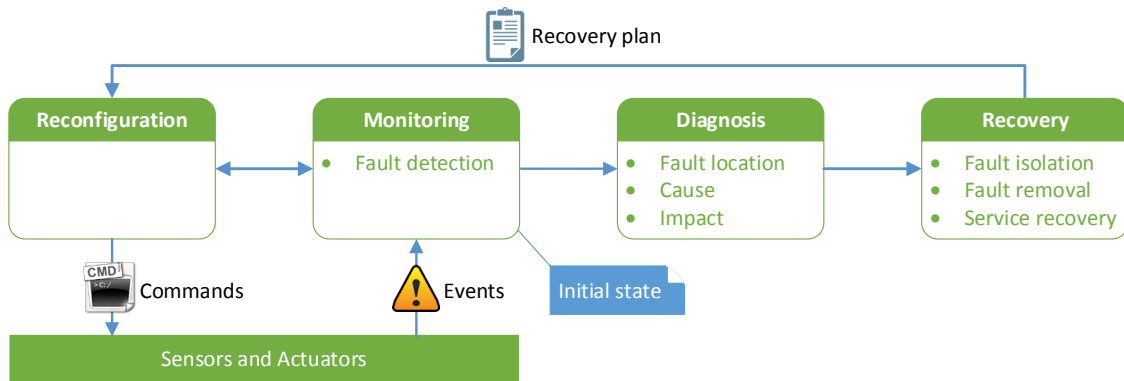


Figure 3.9: Self-healing process model.

- Recovery - Creates a fault removal and/or isolation action sequence - the recovery plan - to recover from occurred failures and to prevent future failures.
 - Fault removal - In fault removal, a set of configurations is applied to the domain components, removing the cause of failure. It is possible only when the required resources are available.
 - Fault isolation - In fault isolation, a set of configurations is applied to the domain components, containing the impact of the failure to the domain or to a subset of its components. It might be employed as a preliminary action to support removal or when it is not possible to completely remove the fault, requiring human intervention.
- Reconfiguration - Smart grid configuration plans are executed by the command center. Therefore, we decided to replace the typical Dispatch supervision module by a Reconfiguration module, which executes only the sequence of actions proposed in the self-healing recovery plan.
 - What happens to the smart grid? - During the several steps of "reconfiguration", the smart grid might go through the "not operational", "degraded", "reconfiguring" and "planned configuration" states.
 - What happens to the smart grid dependent processes? - The processes might go through the "no execution", "degraded execution" and "normal execution" states.

Further information about the how the modules perform the aforementioned activities is given in Section 3.7.2.

3.7.2 Functional Description

The functional description sets its focus on the functional modules of the SHEE architecture, including the following.

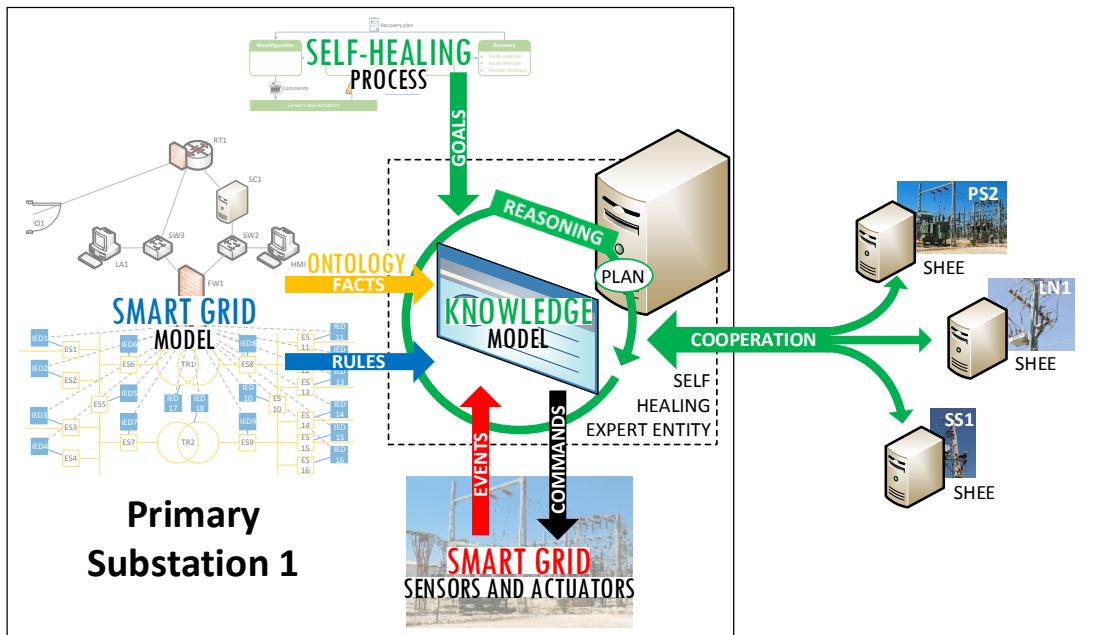
Collectors

There can be different numbers and types of collector modules for different devices and types of devices, which receive and retrieve information from a set of sources and deliver it to the BFT SMR module. The BFT SMR then ensures a total order update of the state of the systems and components services. Regarding the source, the information can be obtained through direct communications with the systems and components (e.g., sensors and actuators, SCADA control devices, network, hosting and security equipment), or indirectly, through data and information concentrators (e.g., SCADA server, network, systems and security monitors). The protocol must be secure. Some components might be configured to send the necessary information on the given schedule, while others must be requested to send this information.

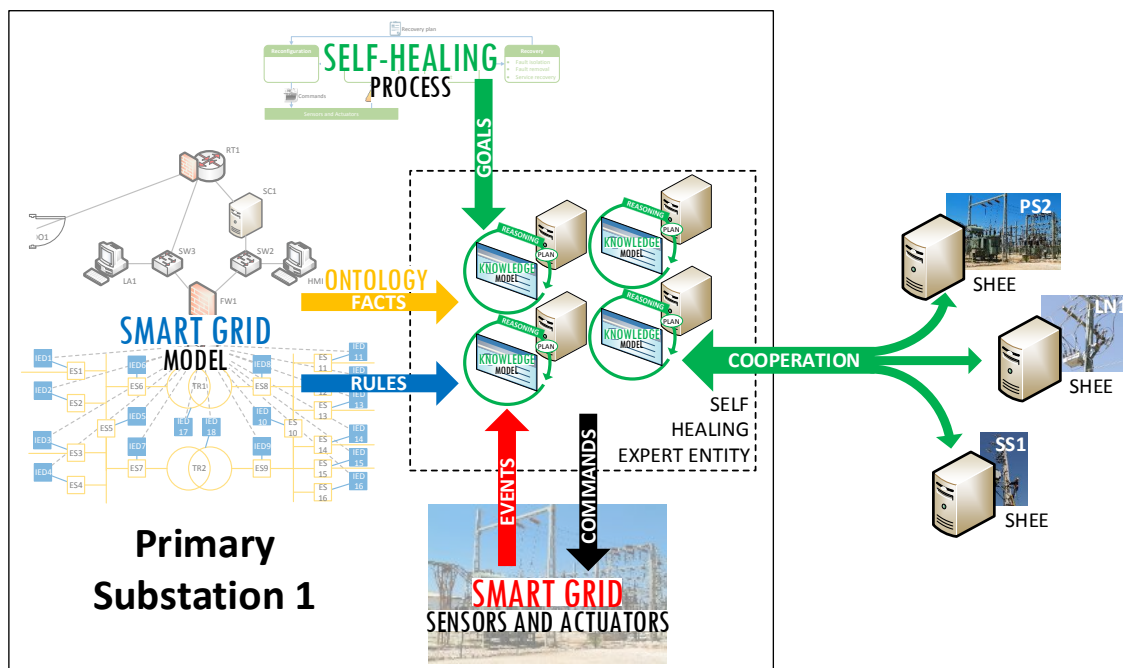
Knowledge

Smart grid information about the systems and components in a given self-healing domain is modeled into facts and rules, which are provided to the knowledge model. As depicted in Figure 3.10, this information includes:

- Facts and rules loaded by the user;
 - The ID of the SHEE;
 - What are the systems and components;
 - What services they offer;
 - How they behave to offer those services;
 - How they should be connected to one-another;
 - How they connect to other self-healing domains (i.e., information about other SHEE);
 - How to update the state of those services;
- Facts and rules acquired by the SHEE:
 - What is the state of those services (i.e., events and alarms), which is provided by the collectors module;
 - Reliabilities of configuration options, which is provided by the learning module.



(a) As perceived by the smart grid.



(b) As perceived by the SHEE.

Figure 3.10: SHEE knowledge and reasoning.

The knowledge module updates this knowledge with the information batch received from the BFT SMR module, triggering the creation of a static copy of the knowledge model and a reaction from the monitoring module. Thereafter, the knowledge module reasons with the copy, through an inference engine, based on goals and queries requested by the other modules. The reasoning results are returned to the requesters, such as recovery plans requested by the recovery module. Each new batch triggers the creation of a new copy and the corresponding reaction from the monitoring module. If the time stamp of new information is older than that of the last information that was considered for the corresponding facts, it is disregarded by the original model. In these situations, the knowledge module creates a new version of the knowledge model copy where the delayed information would have been inserted and includes it there.

The knowledge model is a limited representation of reality. Therefore, the closer it is to the real world, the better and safer can be the self-healing results.

Monitoring

The monitoring module submits goals⁷ to the knowledge module to validate if the supervised self-healing domain's systems and components are properly configured to provide the supported services and enable the dependent processes. The detected faults and failures are subject to diagnosis. It also validates the information obtained by the collectors through its comparison with redundant information from different sources. If an inconsistency is detected, a warning is posted in the user interface and the inconsistent information is not used in the self-healing process, until the inconsistency has been solved. It is possible to monitor also other aspects of its execution through the user interface.

Diagnosis

The diagnosis module submits queries to the knowledge module to determine the cause and impact of observed failures and to find possible causes for future failures. The cause is a set of faulty configurations in the smart grid systems or components. The impact comprises to which extent one or more services were affected directly, indirectly and the number of clients affected. If the affected service is related with other self-healing domains, the corresponding SHEEs are asked to cooperate in the diagnosis. It is possible to monitor its execution through the user interface.

Recovery

The recovery module submits queries to the knowledge model to create a smart grid re-configuration plan to isolate and/or remove the fault and recover the affected services and,

⁷A goal is a proposition that the knowledge module classifies as being true or false.

consequently, the affected smart grid dependent processes. If the affected service is related with other self-healing domains, the corresponding SHEEs are asked to cooperate in the recovery. One of the possible outputs might be that nothing can be done by the SHEE, depending on the available resources. In this situation, the operators may still use the information provided by the diagnosis module. The recovery plan is delivered to the reconfiguration and learning modules. It is possible to monitor its execution through the user interface.

Reconfiguration

The reconfiguration module executes the recovery plan, one step at a time. The reconfiguration messages are identified by a time stamp, a recovery plan identification number and a sequence number that are validated by the controlled devices. If the communication fails, it executes a user configurable number of retries. If none of the retries is successful, it interrupts the reconfiguration and resumes to the other activities. Additionally, as the applicability of the recovery plan is limited by the representativity of the knowledge model, the results of its execution may deviate to an undetermined extent from the expected results. In this context, as each step of the plan is executed, the systems and components generate new events and alarms, as they would normally do, which are monitored and used to update the knowledge model. If the smart grid assumes an unexpected configuration between execution steps, the execution stops as the diagnosis activities are resumed. It is possible to monitor its execution through the user interface. The following alternative behaviors can be observed:

- Proposal-only mode - The recovery plan is presented in the user interface and the reconfiguration terminates.
- Give priority to the operator's actions - The recovery plan is presented in the user interface and it waits for the operator's validation. If there is new and related information during the waiting period, the proposal is withdrawn as the diagnosis activities are resumed.

The reconfiguration steps must be logged, including which component, which configuration, when and with which result, regarding the success or failure of the communication.

Controllers

There can be different numbers and types of controller modules for the various devices. They send reconfiguration commands to the actuators within a self-healing domain, validating their reception within a user configurable time interval. The results are reported to the reconfiguration module. They can send the commands through direct communications with the systems and components (e.g., SCADA control devices, network, hosting and

security equipment), or indirectly, through existing smart grid controllers (e.g., SCADA server), in a similar manner to what was mentioned for the collectors. In both situations, the controllers must use secure protocols.

Learning

The smart grid system or component's configuration possibilities can be associated with a calculated reliability value, which reflect the successes and failures of its use in recovery plans. The learning module compares the expected results with the obtained results to calculate and assign a new reliability value. All changes must be logged. The knowledge model rules use this value to prioritize different reconfiguration sequences, when queried by the recovery module.

Cooperation

A SHEE will contact or be contacted by other SHEEs in cases when the resources within one's self-healing domain, including the knowledge or control capabilities, are not sufficient to perform or complete a thorough diagnosis, recovery planning and/or reconfiguration. This situation occurs when a one or more related faults or the cause and/or impact of a failure involves more than one self-healing domain or when a SHEE's resources are degraded or unavailable. For a better understanding, consider the following examples:

Example 1 Consider a SHEE that supervises the local systems in a primary substation and another SHEE that controls that supervises the central systems perimeter security. A malicious access from the central systems to the substation can be stopped by adding rules to only one or to both firewalls. However, a reconfiguration on only the substation firewall may leave other substations and automation cabinets vulnerable, while the reconfiguration of the central firewall does not prevent the malicious access to the substation if it instead comes from within the substation WAN. Considering that the access is detected by the substation SHEE, it will contact the second SHEE and they will cooperate to create and execute the optimal recovery plan based on to their knowledge.

Example 2 Consider a third SHEE that supervises the local systems in another primary substation, in addition to the SHEEs presented in Example 1. The malicious access caused a fault in an IED in the first substation that resulted in an outage and in the inoperability of the corresponding MV circuit breaker. Without the possibility to recover the service by itself, the affected substation SHEE contacts the third SHEE, which has also electrical connections to the affected electrical network sections and sufficient electrical power to cover the corresponding electricity demand. They cooperate to create and execute the optimal recovery plan based on their knowledge.

If there are several alternatives for cooperation, the SHEE will negotiate with the corresponding SHEEs the creation and execution of the optimal recovery plan. There cannot be common systems or components between any two self-healing domains to prevent concurrency issues during reconfigurations. When a SHEE contacts another, each replica sends to the second a set of fault and failure related facts. The second SHEE replicas know which other SHEE replicas can request contact. Through a secure protocol, the cooperation module at each second SHEE replica authenticates the requesters and waits until a sufficient number of requests has been received. It also compares the requests with one-another to check that they are the same and eventually disregard the requests that act different from the majority. It submits queries to the knowledge module to assess the available resources during negotiations. It delivers knowledge facts updates to the BFT SMR module to reflect the received information in the knowledge module, triggering the self-healing process internally.

Information Sources

The sources know which SHEE replicas require their information, how and on what schedule. They connect to the SHEE replicas to send the required information, through secure protocols. They handle information to the SHEE replicas only when it has been requested by a sufficient number of authenticated replicas, also through secure protocols.

Controlled Devices

The controlled devices know which SHEE replicas require control over them. They apply the controls requested by the SHEE replicas only when it has been requested by a sufficient number of authenticated replicas, through secure protocols. They validate the command time stamp, recovery plan identification number and sequence number. The received time stamps must be within a user configurable time interval, otherwise they are discarded. They execute the messages corresponding to each plan sequentially and in order. They execute the plans in order, discarding any non-executed messages from a previous plan, upon the reception of commands from a new plan.

User Interface

The user interface module supports local and remote management, operation and maintenance of a SHEE replica, enabling access and control over the knowledge, goals, queries, users and related information, logs, cryptographic material and configurations, depending on the specific user access profile. It allows monitoring of the SHEE activities and selection of the mode of operation.

3.7.3 Non-functional Description

The non-functional description sets its focus on the functional modules of the non-functional architecture, including the following.

SMR

The SMR module provides the BFT SMR service to the SHEE, by defining the ordered batch of knowledge updates that are submitted to the knowledge module of each SHEE replica at each time, based on the inputs provided by the collectors, cooperation and learning modules.

Users

The users module enables addition, editing and removal of local users and directory service profiles, providing local and enabling centralized authentication and authorization. There is the possibility to enable fallback to the local credentials, when the directory service is unavailable. The local users and directory service profiles can be associated with different roles and resources.

Cryptography

The cryptography module provides cryptographic functions to other modules and handles the corresponding cryptographic material, such as keys, communicating with specific hardware when required. For example, the knowledge, goal and query and other SHEE related information stored to disk is encrypted. The used secure protocols also encrypt the communications with the information sources and controllers and between SHEE.

Backup

The backup module provides on-demand and scheduled backup of the knowledge, goals, queries, users and related information, logs, cryptographic material and configurations of the other modules.

Configuration

The configuration module allows the remote deployment of configurations and updates to the other modules, including knowledge, goals, queries, backup schedules, addition, editing and removal of users and related parameters, collectors, controllers, cryptographic functions and material.

Logging

The logging module logs the activities performed by the other modules, including changes to the knowledge module, goals and queries, goal verification and query results, issued commands, deployment of configurations, execution of backups and contacts by other SHEE.

Chapter 4

Implementation

This chapter describes a POC implementation of the knowledge related components of the SHEE, including its validation and a discussion of the overall solution. It starts with the selection of the self-healing domain, including its systems, components and services, creation of the corresponding knowledge model, from the ontology to the facts and rules, and the definition of the monitoring, diagnosis and recovery goals and queries. Then, it supports the research hypothesis and validates the SHS concept as a solution to the research problem, defining the validation methodology, the input knowledge for three complexity increasing scenarios and presenting the corresponding results and analysis. It also validates the solution robustness. Finally, it discusses a set of design and implementation issues that, being critical to the security and robustness of the SHS, depend on each smart grid specific context.

4.1 Chapter Overview

A SHEE is defined by the assigned self-healing domain, the monitoring capabilities, the knowledge, the goals and queries and the control capabilities. In this regard, the POC implementation includes the following steps:

1. The selection of the self-healing domain, which comprises a set of primary substation components from the electrical, the control, the communications, the physical security and the cyber security layers, which support the electricity distribution process and the local and remote operation of the substation. This set was chosen for its diversity in terms of included smart grid processes and architecture layers, which allows to focus our efforts on a well-delimited test case, while allowing to draw conclusions that expectedly apply also to other domains that contain the same set or a subset of the same processes and layers.
2. The components are mapped to classes, properties and individuals of an ontology, from which we extract knowledge facts to create the knowledge model. The facts

reflect the existing relations between the ontology individuals (i.e., the domain components) from the moment when they are asserted to the knowledge model until the moment when they are removed.

3. The SHEE attempts to solve a connectivity problem, attending to the cyber-physical connectivity that enables the availability of the electricity distribution process and of the local and remote operation of the substation. For this purpose, we define and assert reasoning rules that enable iterations over the domain components to verify the existence of connectivity enabling operational conditions, while modeling their behavior.
4. We define the goals and queries which are used by the monitoring, diagnosis, recovery, reconfiguration, cooperation and learning agent behaviors to query the knowledge model.

In the following, we validate the implementation, giving evidence that the hypotheses of Section 3.2.3 holds in practice and that the SHS concept is a solution to the research problem. For this purpose, we make three slightly modified versions of the developed knowledge model, corresponding to different fault and failure scenarios. The models are loaded into a standalone Prolog inference engine - SWI-Prolog [80] - to which the goals and queries are submitted in an ordered sequence, simulating the behavior of a SHEE replica through the different states of the self-healing process for each scenario. In this regard, only the knowledge model code was implemented in the scope of the POC. The software code for the remaining modules is left for future work. The robustness of the proposed solution is validated by explaining how it prevents a set of liveness and safety compromising scenarios.

Still within the scope of the current work, there is a set of design and implementation issues that, being critical to the security and robustness of the SHS, depend on each smart grid specific context. In this regard, we provide guidance for the following:

- The SHS must be integrated with existing electrical network self-healing and security automation solutions, avoiding concurrency situations and fostering synergies that are only possible through their integration.
- It can take advantage of the existing smart grid infrastructure, namely, of the existing communications infrastructure, information repositories and smart grid controllers, avoiding the deployment of a self-healing specific communications infrastructure and the direct connection with the sensors and actuators, which may require the implementation of certain safety measures.
- It might be possible to implement a SHEE starting from the existing electrical network self-healing and security automation solutions, which demonstrate some of

the supervision behaviors required by the SHEE modules and are already capable of collecting the required information and controlling certain smart grid components.

- It is limited in its decisions and actions by the available resources, which include the monitoring and control capabilities, the knowledge and the goals and queries. Therefore, the design and implementation of the required collectors and controllers, the use of secure and dependable communication channels and the definition of an enabling knowledge representation, populated with the required facts, rules and queries, are essential to a successful execution of the self-healing process.
- Its correct execution depends on its distribution granularity, on an adequate number of replicas per SHEE and on a proper replica distribution. Therefore, these steps should ensure an acceptable risk to the smart grid network locations, regarding partitioning, and to the unique points of failure between SHEE replicas.

The proper design and implementation decisions contribute to a successful and correct execution of the self-healing process by the SHS. They also provide relevant information regarding the cause and impact of failures that might be used to reduce incident response times and to prevent incident recurrences.

4.2 Proof of Concept

For proof of concept, we make an initial implementation of a SHEE - SHEE1 - focusing mainly on its knowledge model and reasoning capabilities. It comprises the selection of a representative self-healing domain, the modeling of the domain's components and their behavior, the definition of the corresponding facts and rules for the knowledge base and the definition of the goals and queries for the monitoring, diagnosis and recovery activities.

4.2.1 Self-healing Domain

The self-healing domain contains the systems, components and services of a primary substation, as depicted in Figure 4.1. It is a subset of the smart grid components previously depicted in Figure 3.3. In the current implementation, we will use: the electrical switch (ES9), the corresponding IED (IED9), the substation controller (SC1), the HMI (HMI1), two switches (SW1 and SW2), the firewall (FW1), the perimeter router with integrated firewall (RT1), the physical access control system (DO1), the electricity distribution process, the local and remote operation of the substation.

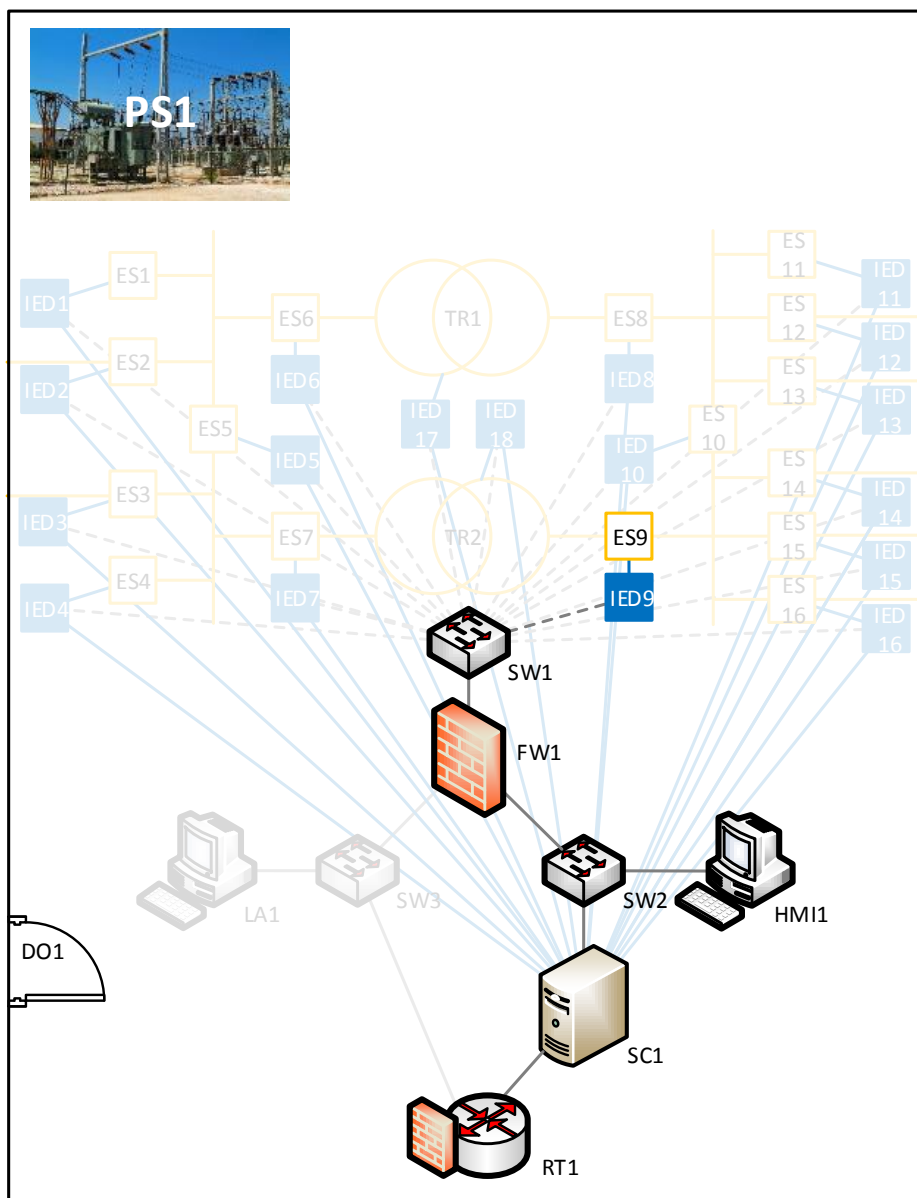


Figure 4.1: Self-healing domain.

4.2.2 Knowledge Modeling

SHEE1 supervises the physical and communications connectivity between the components in its self-healing domain. In this regard, it assumes that this connectivity is required to ensure the availability of the SCADA service for the local operation and remote operation of the substation. It supervises the electrical connectivity, which is required for the electricity distribution process. It also supervises the compliance of the current configurations with the planned configurations. For instance, when an electrical switch is closed, it is allowing the passing of electricity through the circuit. Similarly, when a firewall has an allow rule for any given source and target, it is allowing communication between the two endpoints. Assuming that the switch is open and that the source of the communication is Substation Controller 1 and the target is IED9. If the firewall rule is illegitimately removed and assuming that the default is to deny all communications, it will not be possible to control the IED to close the switch. The created recovery plans comply with the planned configurations and contribute to the compliance of the components' configurations. Therefore, a fault in the firewall configurations caused a connectivity failure, which caused a failure in the SCADA service with impact in the electricity distribution process.

In this context, we define a set of knowledge facts that represent the domain components' information, including their configurations, connections and relations. We define a set of rules that model the components' behaviors and enable iterations over the components' information. We also define the goals and queries that use these rules to verify the existence of connectivity enabling configurations, to diagnose configuration faults and to create recovery plans. In the following sections, we use a few examples to further explain the defined knowledge facts and rules. The Appendix B contains the complete implemented knowledge base that we use in the functional validation.

Component Description

A detailed representation of the self-healing components is depicted in Figure 4.2, including the modeled interfaces, internal and external connections. For example, the router with firewall RT1 comprises the following interfaces and connections.

- Ethernet interface 2 (ETH2) - It is connected with the Core interface, supporting communications from the outside to the router core. The enabled and allowed communications depend on the configured routes and firewall rules, respectively. A firewall rule must be configured in an interface and contain a source, a target and a permission, the last being allowed or denied ¹. A route must also be configured in an interface and indicate that the source or target host is behind that interface. In

¹It is assumed that RT1 implements a stateful firewall, allowing also the replies that match a corresponding established session.

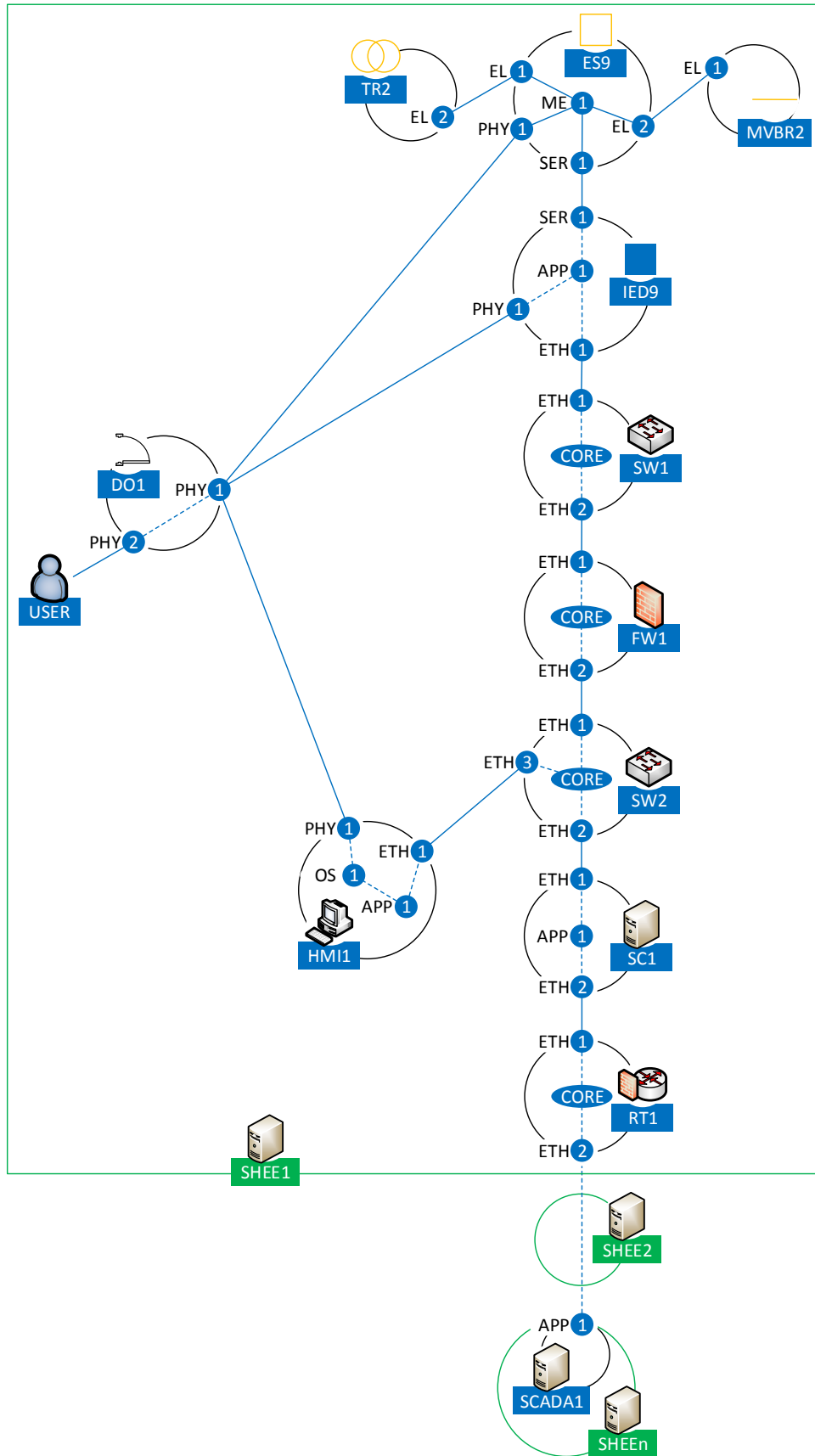


Figure 4.2: Detailed self-healing domain components.

this regard, it has one route and one firewall rule configured, enabling the communications back to SCADA1 and allowing communications between SCADA1 and SC1.

- Core interface (CORE) - It represents the router core, being connected with the Ethernet Interface 1 and supporting the communications between the several interfaces.
- Ethernet interface 1 (ETH1) - It enables and allows communications, depending on the configured routes and firewall rules, respectively. In this regard, it has one route and one firewall rule configured, enabling the communications forward to SC1 and allowing communications between SCADA1 and SC1.

The firewall configurations, for example, are verified and validated according to the following reasoning rules:

- Required configuration - To allow communications between a source and a target hosts in a certain interface, it must be configured with a firewall rule that allows communications between both.
- Monitoring - The communications are allowed if the current state reflects the required rule.
- Diagnosis - If there is a rule denying the required communications or if there are no configured rules, then that is a faulty configuration that is contributing the communications failure.
- Recovery - If there is a rule with the wrong permission, then it must be changed. If there are no configured rules, then a new rule must be configured to allow communications between the required source and target hosts.

SCADA1 is a special component in relation to the remaining (see bottom of Figure 4.2). It is not monitored by SHEE1 but the latter knows of its existence through SHEE2. SCADA1 is contained in SHEE2's self-healing domain or in a domain previous to that. Therefore, there must be cooperation between at least the two aforementioned SHEEs during the diagnosis and recovery activities.

In addition to firewall and routing, the implementation includes a set of controls related with switching, authentication, operating modes and electrical switch position. For some components, such as SC1, HMI1, IED9, ES9 and DO1, we consider the existence of planned configurations or policies for the aforementioned controls. For instance, the planned configurations might require that a certain set of local users is configured in the operating system or the application, which is the case of HMI1 and DO1. It might require the setting of a certain operating mode, as in SC1 and IED9, where, in the latter case, it is not controllable by the SHEE. It might also require the ES9 to be in a certain state. In

this context, the authentication configurations are verified and validated according to the following reasoning rules:

- Required configuration - To allow the access of a user to an interface. Then, that user must be configured in that interface.
- Monitoring - The access is and should be allowed if the current state reflects the required configuration and it complies with the planned configuration.
- Diagnosis - The current configurations are faulty in the following scenarios.
 1. The current configurations reflect the required configuration. However, they are not compliant with the planned configurations.
 2. The current configurations do not reflect the required configuration which complies with the planned configuration.
 3. The current configurations comply with the planned configurations but neither complies with the required configuration.
- Recovery - In scenario 1, the user must be removed from the interface configurations. In scenario 2, it must be configured. In scenario 3, the user cannot be configured because it does not comply with the planned configurations, which should generate an alarm.

Facts

The set of facts corresponding to the RT1 description is depicted in Figure B.1, written in the Prolog programming language [81]. A string with the structure $\langle predicate \rangle (\langle atom1 \rangle, \langle atom2 \rangle, \langle atom3 \rangle)$ is called a clause. A single clause is a fact. The first fact says that RT1 is a component of the router type. The second to fourth facts say that it has three interfaces. The first is of the core type, while the others are of the ethernet type. The fifth to ninth facts refer to the connections between the interfaces. The tenth and eleventh fact say that in a firewall rule deny and allow have opposing connotations. These support the communications, depending on the firewall and routing configurations at each interface. The last facts define the current state, including the configured routes and firewall rules.

Rules

The set of rules corresponding to firewall and authentication are depicted in Figures B.13 and B.14, also written in the Prolog programming language. There are monitoring, diagnosis and recovery rules defined for each control. A string with the structure $\langle clause1 \rangle: - \langle clause2 \rangle, \langle clause3 \rangle, \langle clause_n \rangle$ is a rule. For example,

the second rule of the Figure B.13 defines that the firewall configuration is correct if the monitored interface contains a rule with the right parameters in its current state. The corresponding rule in Figure B.14 defines that the user configuration is correct if the required configuration (to allow a given user access) is allowed by the planned configuration, here represented by the policy predicate, and that user is configured in the monitored interface.

The rules in Figures B.18 to B.24 allow recursive iterations over the previous facts and rules, defining each self-healing activity's work flow. For instance, in a similar way, the monitoring and diagnosis recursive rules iterate over the monitoring and diagnosis rules that are specific to each component, validating the configurations and, in the second case, identifying faulty configurations. The diagnosis creates a diagnostic - a list of configurations with indication of the reason why they did not pass the validation. This list is submitted to the recovery, which iterates over the configurations and employs the components specific recovery rules to assign reconfigurations. The result is a reconfiguration plan that would be submitted to the reconfiguration module.

4.2.3 Goals and Queries

The knowledge facts and rules are loaded into an inference engine so that they can be queried by a user or, in this case, by the SHEE modules. A query has the same structure as a fact. However, the values of some atoms might not be defined, requiring the inference engine to deduce them. In the chosen self-healing domain, the SHEE supervises the electrical connectivity between the electrical network components, the communications, the physical access to the substation and the SCADA service that supports the local and remote operations. Its goals are to detect and heal faulty configurations, preventing and recovering from service failures. Therefore, we define the following queries for the monitoring, diagnosis and recovery activities:

- Monitoring:
 - *monitor(scada, scada1 : app1, es9 : el2, _)*. - Monitors the SCADA service, focusing on communications originated at SCADA1 that have impact in the electrical connectivity at ES9.
 - *monitor(scada, do1 : phy2, es9 : el2, _)*. - Monitors the SCADA service, considering a physical access from the outside of DO1 with impact in the electrical connectivity at ES9.
- Diagnosis:
 - *diagnose(scada, scada1 : app1, es9 : el2, _, DiagnosisResult)*. - Diagnosis the configurations required by the SCADA service, focusing on communications originated at SCADA1 that have impact in the electrical connectivity at ES9.

- *diagnose(scada, do1 : phy2, es9 : el2, -, DiagnosisResult)*. - Diagnosis the configurations required by the SCADA service, considering a physical access from the outside of DO1 with impact in the electrical connectivity at ES9.
- Recovery:
 - *recover(scada, scada1 : app1, es9 : el2, -, RecoveryPlan)*. - Returns the plan required to recover the SCADA service, focusing on communications originated at SCADA1 that have impact in the electrical connectivity at ES9.
 - *recover(scada, do1 : phy2, es9 : el2, -, RecoveryPlan)*. - Returns the plan required to recover the SCADA service, considering a physical access from the outside of DO1 with impact in the electrical connectivity at ES9.

It is also possible to query the communications, electrical and physical layers directly by replacing the "scada" atom with, for example, "electrical". Moreover, we use a "_" in the queries because we do not require the rules to consider a specific user. However, if a specific user must be supervised, the corresponding atom can be replaced with, for example, "alice".

4.3 Functional Validation

The functional validation targets the initial implementation of the knowledge base, the goals and the queries.

4.3.1 Methodology

The facts and rules were loaded into a standalone Prolog inference engine - SWI Prolog - to which a set of goals and queries were submitted in an ordered sequence, simulating the behavior of the SHEE through the different stages of the self-healing process (see Figure 4.3), for three different fault and/or failure scenarios. Remembering the purpose of each self-healing stage, after a knowledge update (see Figure 4.3a), the monitoring activity will be triggered and the SHEE will validate the components' configurations. In Figure 4.3b, it detects faulty configurations in ES9, which are causing a failure in the electricity distribution process. In the diagnosis activity (see Figure 4.3c), it associates the fault with an unknown user in HMI1. In the recovery activity (see Figure 4.3d), it creates a recovery plan that includes the removal of the unknown user. Other plans could be possible, such as to deny the communications from HMI1 or even SC1 in FW1. The results should be equivalent to using a software agent implementation, which is due to the fact that, by design, the intelligence of a SHEE is embedded in its knowledge model, goals and queries - not in its software code.

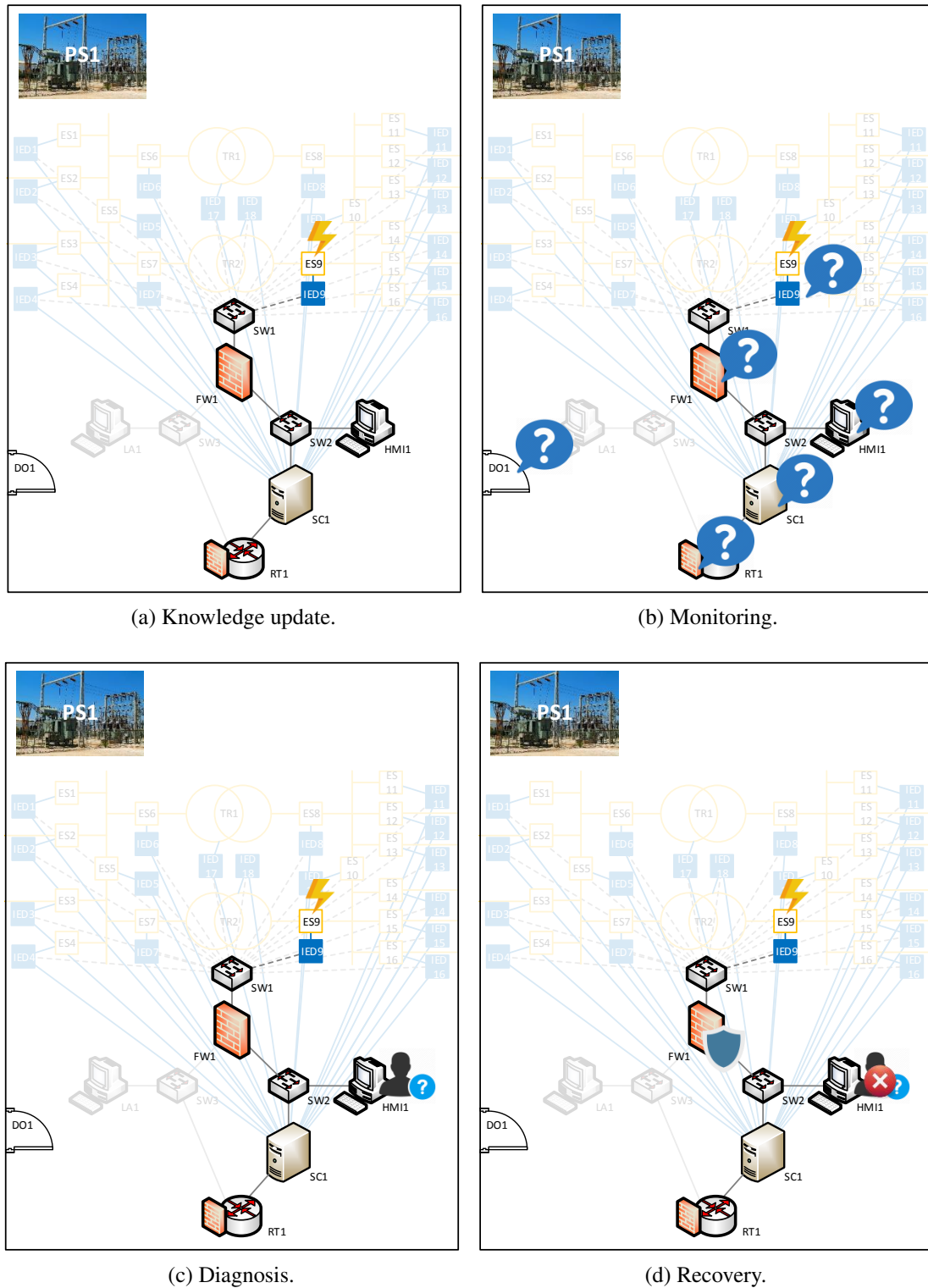


Figure 4.3: Self-healing cycle.

4.3.2 Scenario 1: A Firewall Reconfiguration

PS1 is a new substation that will be inaugurated in a few days by the Blue Planet² DSO. However, they are having problems with the communications to SCADA1. SHEE1 was just activated and it is supervising the corresponding self-healing domain.

Knowledge

The same as in Appendix B, with the following changes. In RT1, we remove the correct rule (with the source *scada1 : app1*) and we add a faulty rule (with the source *scaadaa1 : app1*). In SC1, we change the SC1 mode from local to remote in the current and planned configurations.

- To the RT1 firewall rules:
 - Remove *currentState(rt1 : eth1, fwRule, (allow, scada1 : app1, sc1 : app1))*.
 - Add *currentState(rt1 : eth1, fwRule, (allow, scaadaa1 : app1, sc1 : app1))*.
- To the SC1 mode:
 - Remove *policy(sc1 : app1, scmode, (local))*.
 - Remove *currentState(sc1 : app1, scmode, (local))*.
 - Add *policy(sc1 : app1, scmode, (remote))*.
 - Add *currentState(sc1 : app1, scmode, (remote))*.

Results

The results are depicted in Figure 4.4.

Analysis

The SHEE discovers that the current RT1 firewall configurations do not include a rule to allow the traffic between SCADA1 and SC1 (*rt1:eth1, noFwRule, allow, scada1:app1, sc1:app1*). Therefore, it creates a recovery plan, where the aforementioned rule is added to the interface ETH1 (*rt1:eth1, addFwRule, allow, scada1:app1, sc1:app1*). Moreover, as SCADA1 is associated with another SHEE - SHEE2 - it would ask for the cooperation of this SHEE to supervise the path from SCADA1 that is outside of its self-healing domain.

²Blue Planet, Alice, Albert and Bob are fictional entities invented and used for the sole purpose of this work.

```

SWI-Prolog -- c:/Users/Nuno Pereira/Nuno/MSI/Dissertação/Prolog/20160925_kbv15_psl[tests].pl
File Edit Settings Run Debug Help

4 ?- monitor(scada, scada1:app1, es9:e12, _).
false.

5 ?- diagnose(scada, scada1:app1, es9:e12, _, Diagnosis).
Diagnosis = [ (rt1:eth2, okRoute, scada1:app1), (rt1:eth2, okFwRule, allow, scada1:app1, sc1:
app1), (rt1:eth1, noFwRule, allow, scada1:app1, sc1:app1), (rt1:eth1, okRoute, sc1:app1), (sc
1:eth2, okRoute, scada1:app1), (sc1:eth2, okFwRule, allow, scada1:app1, sc1:app1), (sc1:app1,
okScmode, remote), (sc1:eth1, okFwRule, allow, sc1:app1, ied9:app1), (sc1:eth1, okRoute, ied
9:app1), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth2, okRoute, sc1:app1), (fw1:eth2, okFwRule, a
llow, sc1:app1, ied9:app1), (fw1:eth1, okFwRule, allow, sc1:app1, ied9:app1), (fw1:eth1, okRo
ute, ied9:app1), (sw1:eth1, okVlan, sw1:vlan1), (ied9:app1, okIedmode, remote), (es9:me1, pos
itionOk, closed)] ;
false.

6 ?- recover(scada, scada1:app1, es9:e12, _, RecoveryPlan).
RecoveryPlan = [ (rt1:eth1, addFwRule, allow, scada1:app1, sc1:app1)] ;
false.

7 ?-

```

Figure 4.4: Scenario 1 results.

4.3.3 Scenario 2:

The Consequences of an Incomplete Knowledge Base

The recovery plan created by SHEE1 was sufficient to recover the SCADA service. The current scenario occurs immediately after these events. Alice, an installation team member from the SCADA provider company, is attempting to login to HMI1. However, HMI1 is replying that Alice credentials are invalid.

Knowledge

The same as in Scenario 1, with the following changes. In HMI1, we remove the planned configurations related to Alice's user from the application (`hmi1:app1`) and from the operating system (`hmi1:os1`). In DO1, we do the same for the physical access control system (`do1:phy2`).

- To the HMI1 users:
 - Remove *policy(hmi1 : os1, user, (alice))*.
 - Remove *policy(hmi1 : app1, user, (alice))*.
- To the DO1 users:
 - Remove *policy(do1 : phy2, user, (alice))*.

Results

The results are depicted in Figure 4.5.

```

1 ?- monitor(scada, dol:phy2, es9:e12, _).
false.

2 ?- diagnose(scada, dol:phy2, es9:e12, _, Diagnosis).
Diagnosis = [ (dol:phy2, userNAByPol, alice), (hmi1:os1, userNAByPol, alice), (hmi1:appl, use
rNAByPol, alice), (hmi1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (hmi1:eth1, okRoute, sc1
:appl), (sw2:eth2, okVlan, sw2:vlan1), (sc1:eth1, okRoute, ... : ...), (... : ... : ... : ...),
(... : ...)] [write]
Diagnosis = [ (dol:phy2, userNAByPol, alice), (hmi1:os1, userNAByPol, alice), (hmi1:appl, use
rNAByPol, alice), (hmi1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (hmi1:eth1, okRoute, sc1
:appl), (sw2:eth2, okVlan, sw2:vlan1), (sc1:eth1, okRoute, hmi1:appl), (sc1:eth1, okFwRule, a
llow, hmi1:appl, sc1:appl), (sc1:appl, scmodeNAPolNConf, local), (sc1:eth1, okFwRule, allow,
sc1:appl, ied9:appl), (sc1:eth1, okRoute, ied9:appl), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth
2, okRoute, sc1:appl), (fw1:eth2, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okFwRule,
allow, sc1:appl, ied9:appl), (fw1:eth1, okRoute, ied9:appl), (sw1:eth1, okVlan, sw1:vlan1),
(ied9:appl, okIedmode, remote), (es9:me1, positionOk, closed)] ;
Diagnosis = [ (dol:phy2, userNAByPol, alice), (ied9:appl, iedmodeNAPolNConf, local), (es9:me1
, positionOk, closed)] ;
Diagnosis = [ (dol:phy2, userNAByPol, alice), (es9:me1, positionOk, closed)] ;
false.

3 ?- recover(scada, dol:phy2, es9:e12, _, RecoveryPlan).
RecoveryPlan = [ (dol:phy2, removeUser, alice), (hmi1:os1, removeUser, alice), (hmi1:appl, re
moveUser, alice), (sc1:appl, scmodeLimByPolicy)] ;
RecoveryPlan = [ (dol:phy2, removeUser, alice), (ied9:appl, iedmodeLimByPolicy)] ;
RecoveryPlan = [ (dol:phy2, removeUser, alice)] ;
false.

4 ?-

```

Figure 4.5: Scenario 2 results.

Analysis

The installation team activated SHEE1 without updating the planned configurations, which do not include Alice's user in DO1 (*(dol: phy2, userNAByPol, alice)*), neither in HMI1's operating system and application (*(hmi1:os1, userNAByPol, alice)* and *(hmi1:appl, userNAByPol, alice)*). These were already defined considering an ongoing situation in which the installation team has no access to HMI1. Therefore, SHEE1 created a recovery plan in which user Alice is removed from DO1 (*(dol:phy2, removeUser, alice)*) and from HMI1 (*(hmi1:os1, removeUser, alice)* and *(hmi1:appl, removeUser, alice)*). She has no authorization to enter the substation or log in to HMI1. Moreover, if she leaves the substation she will not be able to reenter to access IED9 or ES9. SHEE1 detected also that SC1 is configured to remote mode and this is compliant with the planned configurations (*(sc1:appl, scmodeNAPolNConf, local)* and *(sc1:appl, scmodeLimByPolicy)*). Therefore, even if Alice could log in, she would not be able to switch and maintain SC1 in local mode until the planned configurations are previously changed. SHEE1 creates two more recovery plans, one for each possible physical access path to the domain components.

4.3.4 Scenario 3: Physical Security and SCADA

The installation team updates the planned configurations to restore Alice's access and add Albert's access, which are required to finish HMI1 installation. The SC1 mode planned configuration is also changed to local to permit the local operation of the substation through HMI1.

Knowledge

The same as in Scenario 2, with the following changes. In HMI1, we remove Alice's user from the current configurations but we add it to the planned configurations, together with Albert's user. We do the same in DO1. In SC1, we change the mode from remote to local in the planned configurations.

- To the HMI1 users:
 - Remove *currentState(hmi1 : app1, user, (alice))*.
 - Remove *currentState(hmi1 : os1, user, (alice))*.
 - Add *policy(hmi1 : app1, user, (alice))*.
 - Add *policy(hmi1 : app1, user, (albert))*.
 - Add *policy(hmi1 : os1, user, (alice))*.
 - Add *policy(hmi1 : os1, user, (albert))*.
- To the SC1 mode:
 - Remove *policy(sc1 : app1, scmode, (remote))*.
 - Add *policy(sc1 : app1, scmode, (local))*.
- To the DO1 users:
 - Remove *currentState(do1 : phy2, user, (alice))*.
 - Add *policy(do1 : phy2, user, (alice))*.
 - Add *policy(do1 : phy2, user, (albert))*.

Results

The results are depicted in Figure 4.6.

Analysis

SHEE1 detects that there are users allowed by the planned configurations that are not configured, referring to Alice and Albert in DO1 and HMI1's operating system and application ((*do1 : phy2, userNotConfigured, alice*), (*hmi1:os1, userNotConfigured, alice*), (*hmi1:app1, userNotConfigured, alice*), (*do1 : phy2, userNotConfigured, albert*), (*hmi1:os1, userNotConfigured, albert*), (*hmi1:app1, userNotConfigured, albert*)). It also detects that the planned configurations permit the SC1 mode to be set to local, which is required for local operation. It is simply not configured in the SC1 application ((*sc1:app1, scmodeNotConfigured, local*)). Therefore, it creates a recovery plan that includes the configuration of these users and the change of SC1 Mode to local. In this regard, Alice's

```

5 ?- monitor(scada, dol:phy2, es9:e12, _).
false.

6 ?- diagnose(scada, dol:phy2, es9:e12, _, Diagnosis).
Diagnosis = [ (dol:phy2, userNotConfigured, alice), (hmi1:os1, userNotConfigured, alice), (hmi1:appl, userNotConfigured, alice), (hmi1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (hmi1:eth1, okRoute, sc1:appl), (sw2:eth2, okVlan, sw2:vlan1), (sc1:eth1, okRoute, hmi1:appl), (sc1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (sc1:appl, scmodeNotConfigured, local, remote), (sc1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (sc1:eth1, okRoute, ied9:appl), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth2, okRoute, sc1:appl), (fw1:eth2, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okRoute, ied9:appl), (sw1:eth1, okVlan, sw1:vlan1), (ied9:appl, okIedmode, remote), (es9:me1, positionOk, closed) ] ;
Diagnosis = [ (dol:phy2, userNotConfigured, albert), (hmi1:os1, userNotConfigured, albert), (hmi1:appl, userNotConfigured, albert), (hmi1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (hmi1:eth1, okRoute, sc1:appl), (sw2:eth2, okVlan, sw2:vlan1), (sc1:eth1, okRoute, hmi1:appl), (sc1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (sc1:appl, scmodeNotConfigured, local, remote), (sc1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (sc1:eth1, okRoute, ied9:appl), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth2, okRoute, sc1:appl), (fw1:eth2, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okRoute, ied9:appl), (sw1:eth1, okVlan, sw1:vlan1), (ied9:appl, okIedmode, remote), (es9:me1, positionOk, closed) ] ;
Diagnosis = [ (dol:phy2, userNotConfigured, alice), (ied9:appl, iedmodeNAPolNConf, local), (es9:me1, positionOk, closed) ] ;
Diagnosis = [ (dol:phy2, userNotConfigured, albert), (ied9:appl, iedmodeNAPolNConf, local), (es9:me1, positionOk, closed) ] ;
Diagnosis = [ (dol:phy2, userNotConfigured, alice), (es9:me1, positionOk, closed) ] ;
Diagnosis = [ (dol:phy2, userNotConfigured, albert), (es9:me1, positionOk, closed) ] ;
false.

7 ?- recover(scada, dol:phy2, es9:e12, _, RecoveryPlan).
RecoveryPlan = [ (dol:phy2, addUser, alice), (hmi1:os1, addUser, alice), (hmi1:appl, addUser, alice), (sc1:appl, changeScmodeTo, local) ] ;
RecoveryPlan = [ (dol:phy2, addUser, albert), (hmi1:os1, addUser, albert), (hmi1:appl, addUser, albert), (sc1:appl, changeScmodeTo, local) ] ;
RecoveryPlan = [ (dol:phy2, addUser, alice), (ied9:appl, iedmodeLimByPolicy) ] ;
RecoveryPlan = [ (dol:phy2, addUser, albert), (ied9:appl, iedmodeLimByPolicy) ] ;
RecoveryPlan = [ (dol:phy2, addUser, alice) ] ;
RecoveryPlan = [ (dol:phy2, addUser, albert) ] ;
false.

8 ?-

```

Figure 4.6: Scenario 3 results.

and Albert's users are added to DO1 and HMI1's operating system and application (*(dol:phy2, addUser, alice)*, *(hmi1:os1, addUser, alice)*, *(hmi1:appl, addUser, alice)*, *(dol:phy2, addUser, albert)*, *(hmi1:os1, addUser, albert)* and *(hmi1:appl, addUser, albert)*). The SC1 mode is set to local (*(sc1:appl, changeScmodeTo, local)*).

4.3.5 Scenario 4:

An Electrical Failure in a Compromised Infrastructure

After the events in Scenario 3, Alice's and Albert's users are removed from the systems and SC1 mode is set to remote, with the correct update of the planned configurations. One month latter, during a stormy night, the operators loose communications to the substation and ES9 is opened without their intervention.

Knowledge

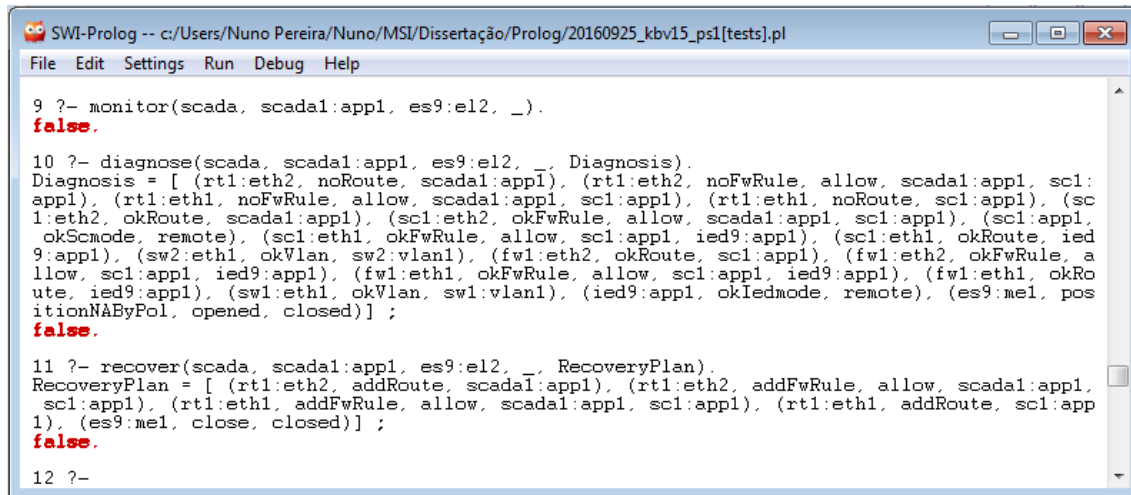
The same as in Scenario 3, with the following changes. In RT1, we remove the routes and the firewall rules from both ethernet interfaces (*rt1:eth1* and *rt1:eth2*). In HMI1, we remove Alice's and Albert's users from the current configurations and we add Bob's

user. These users are also removed from the planned configurations of DO1. In SC1, we change the mode from local to remote. In ES9, we change the current configuration of the mechanism (es9:me1) from closed to open.

- To the RT1 routes and rules:
 - Remove *currentState*(rt1 : eth1, route, (sc1 : app1)).
 - Remove *currentState*(rt1 : eth2, route, (scada1 : app1)).
 - Remove *currentState*(rt1 : eth1, fwRule, (allow, scada1 : app1, sc1 : app1)).
 - Remove *currentState*(rt1 : eth2, fwRule, (allow, scada1 : app1, sc1 : app1)).
- To the HMI1 users:
 - Remove *policy*(hmi1 : app1, user, (alice)).
 - Remove *policy*(hmi1 : os1, user, (alice)).
 - Remove *policy*(hmi1 : app1, user, (albert)).
 - Remove *policy*(hmi1 : os1, user, (albert)).
 - Add *currentState*(hmi1 : app1, user, (bob)).
- To the SC1 mode:
 - Remove *policy*(sc1 : app1, scmode, (local)).
 - Add *policy*(sc1 : app1, scmode, (remote)).
- To the ES9 position:
 - Remove *currentState*(es9 : me1, position, (closed)).
 - Add *currentState*(es9 : me1, position, (opened)).
- To the DO1 users:
 - Remove *policy*(do1 : phy2, user, (alice)).
 - Remove *policy*(do1 : phy2, user, (albert)).

Results

The results are depicted in Figure 4.7.



```

SWI-Prolog -- c:/Users/Nuno Pereira/Nuno/MSI/Dissertação/Prolog/20160925_kbv15_ps1[tests.pl]
File Edit Settings Run Debug Help

9 ?- monitor(scada, scada1:appl, es9:e12, _).
false.

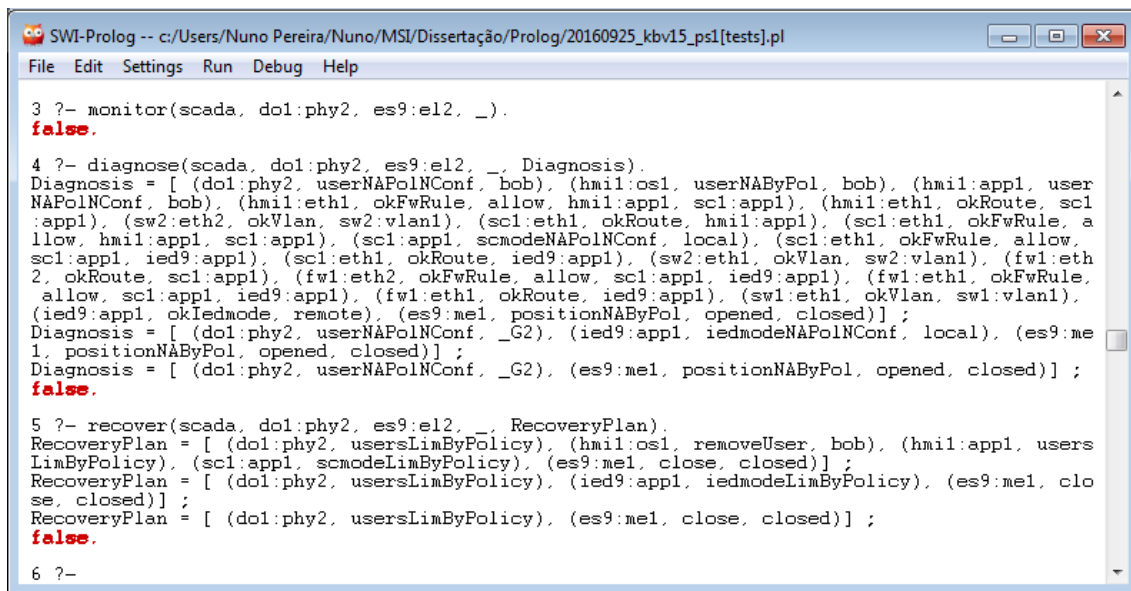
10 ?- diagnose(scada, scada1:appl, es9:e12, _, Diagnosis).
Diagnosis = [ (rt1:eth2, noRoute, scada1:appl), (rt1:eth2, noFwRule, allow, scada1:appl, sc1:appl), (rt1:eth1, noRoute, scada1:appl), (sc1:eth2, okRoute, scada1:appl), (sc1:eth2, okFwRule, allow, scada1:appl, sc1:appl), (sc1:appl, okScmode, remote), (sc1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (sc1:eth1, okRoute, ied9:appl), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth2, okRoute, sc1:appl), (fw1:eth2, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okRoute, ied9:appl), (sw1:eth1, okVlan, sw1:vlan1), (ied9:appl, okIedmode, remote), (es9:me1, positionNAByPol, opened, closed)] ;
false.

11 ?- recover(scada, scada1:appl, es9:e12, _, RecoveryPlan).
RecoveryPlan = [ (rt1:eth2, addRoute, scada1:appl), (rt1:eth2, addFwRule, allow, scada1:appl, sc1:appl), (rt1:eth1, addFwRule, allow, scada1:appl, sc1:appl), (rt1:eth1, addRoute, sc1:appl), (es9:me1, close, closed)] ;
false.

12 ?-

```

(a) Scenario 4 results regarding the remote operation.



```

SWI-Prolog -- c:/Users/Nuno Pereira/Nuno/MSI/Dissertação/Prolog/20160925_kbv15_ps1[tests.pl]
File Edit Settings Run Debug Help

3 ?- monitor(scada, dol:phy2, es9:e12, _).
false.

4 ?- diagnose(scada, dol:phy2, es9:e12, _, Diagnosis).
Diagnosis = [ (dol:phy2, userNAPolNConf, bob), (hmi1:os1, userNAByPol, bob), (hmi1:appl, userNAPolNConf, bob), (hmi1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (hmi1:eth1, okRoute, sc1:appl), (sw2:eth2, okVlan, sw2:vlan1), (sc1:eth1, okRoute, hmi1:appl), (sc1:eth1, okFwRule, allow, hmi1:appl, sc1:appl), (sc1:appl, scmodeNAPolNConf, local), (sc1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (sc1:eth1, okRoute, ied9:appl), (sw2:eth1, okVlan, sw2:vlan1), (fw1:eth2, okRoute, sc1:appl), (fw1:eth2, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okFwRule, allow, sc1:appl, ied9:appl), (fw1:eth1, okRoute, ied9:appl), (sw1:eth1, okVlan, sw1:vlan1), (ied9:appl, okIedmode, remote), (es9:me1, positionNAByPol, opened, closed)] ;
Diagnosis = [ (dol:phy2, userNAPolNConf, _G2), (ied9:appl, iedmodeNAPolNConf, local), (es9:me1, positionNAByPol, opened, closed)] ;
Diagnosis = [ (dol:phy2, userNAPolNConf, _G2), (es9:me1, positionNAByPol, opened, closed)] ;
false.

5 ?- recover(scada, dol:phy2, es9:e12, _, RecoveryPlan).
RecoveryPlan = [ (dol:phy2, usersLimByPolicy), (hmi1:os1, removeUser, bob), (hmi1:appl, usersLimByPolicy), (sc1:appl, scmodeLimByPolicy), (es9:me1, close, closed)] ;
RecoveryPlan = [ (dol:phy2, usersLimByPolicy), (ied9:appl, iedmodeLimByPolicy), (es9:me1, close, closed)] ;
RecoveryPlan = [ (dol:phy2, usersLimByPolicy), (es9:me1, close, closed)] ;
false.

6 ?-

```

(b) Scenario 4 results regarding the local operation.

Figure 4.7: Scenario 4 results.

Analysis

In this scenario, SHEE1 detected the opened ES9, which is not in compliance with the planned configurations (*(es9:me1, positionNAByPol, opened, closed)*), causing a failure in the electricity distribution process. It also detects an unknown user in the operating system of HMI1 - Bob - and faulty firewall and routing configurations in RT1. Bob is not known in the planned configurations (*(hmi1:os1, userNAByPol, bob)*). There are no routes in RT1's ETH2 and ETH1 to enable communications back to SCADA1 and forward to SC1, respectively (*(rt1:eth2, noRoute, scada1:appl)* and *(rt1:eth1, noRoute, sc1:appl)*). There are no firewall rules at the same interfaces to allow communications between SCADA1 and SC1 (*(rt1:eth2, noFwRule, allow, scada1:appl, sc1:appl)* and

(rt1:eth1, noFwRule, allow, scada1:app1, sc1:app1)). Therefore, it creates a recovery plan with the necessary configurations. Bob is removed from HMI1's operating system (*(hmi1:os1, removeUser, bob)*). Routes are added to RT1's ETH2 and ETH1 to enable the aforementioned communications (*(rt1:eth2, addRoute, scada1:app1)*), (*(rt1:eth1, addRoute, sc1:app1)*). The same happens with the firewall rules (*(rt1:eth2, addFwRule, allow, scada1:app1, sc1:app1)*) and (*(rt1:eth1, addFwRule, allow, scada1:app1, sc1:app1)*). ES9 is also closed (*(es9:me1, close, closed)*). Like in Scenario 1, it would ask for the cooperation of SHEE2 to supervise the path from SCADA1 that is outside of its self-healing domain. With just the current knowledge, SHEE1 is not capable to determine the origin of the intrusion. If it happens again, SHEE1 will create a new recovery plan and reapply the configurations to the self-healing domain components. Moreover, in the possible event of an unsuccessful ES9 reconfiguration, it would consider alternative plans, including asking for the cooperation of other SHEEs supervising substations that are connected to its own.

4.4 Security Validation

A set of security requirements was employed in the design of the SHS. These requirements were obtained through a threat and vulnerability assessment and identification of security and dependability controls. A robust implementation of these controls provides the required protection to the SHS.

4.5 Robustness Validation

To validate the robustness of the proposed solution, we analyze a set of scenarios in which the following definitions of liveness and safety are compromised and we provide an explanation regarding how they are prevented.

- Liveness - The SHS should execute continuously from its setup to the end of its life cycle, between programmed interruptions.
- Safety - The SHS's actions and/or proposals are based only on up-to-date information collected from the smart grid systems and components.

4.5.1 Liveness Validation

The liveness validation focus on the interactions between modules, with the smart grid systems and components and with other SHEE.

The collectors stop collecting information.

The collectors are triggered by connections from the smart grid information sources and by scheduled events. By design, they will only stop collecting information if the sources

stay silent or if there are no scheduled events. Both situations can be prevented by adequate setup and maintenance of the SHS.

The SMR module halts indefinitely.

Assuming that at least $n - f$ replicas are available, the BFT SMR protocol executes continuously within the SMR module and between the different replicas.

The self-healing process is stopped between cycles.

The monitor is triggered by the knowledge module updates. Therefore, it will not be triggered if there are no knowledge updates. Assuming that the collectors are working correctly and with at most f faulty replicas, there being no knowledge updates means that the smart grid state stays unchanged since the previous update. This is an acceptable situation. Regarding the diagnosis, recovery and reconfiguration activities, each is triggered by the ending of the corresponding previous activity.

The reconfiguration halts indefinitely when controlling the smart grid.

At each reconfiguration step, the reconfiguration module asks the controllers to send a command to a certain smart grid system or component. It then waits during a user configurable time period for a confirmation that the command was successfully delivered. If it receives a confirmation, it proceeds to confirm the reconfiguration success or failure in the knowledge base. If the reconfiguration was successful, it resumes to the next reconfiguration step. If it receives no confirmation, it retries for a user configurable number of tries. If there is still no delivery confirmation or if the reconfiguration result is different from what was expected, it resumes with the remaining activities.

A self-healing activity halts indefinitely when asking for cooperation.

A SHEE requires cooperation from other SHEE when its self-healing domain resources are not sufficient to perform a thorough diagnosis, recovery planning and/or reconfiguration. In this scenario, it requests cooperation from the SHEEs in contiguous self-healing domains, sending to them information regarding the fault and/or failure. Then, it waits for diagnosis information or recovery/reconfiguration proposals during a user configurable time period. Upon receiving a sufficient number of consistent messages from the corresponding replicas for each information or proposal within a user configurable time period, it uses the information for diagnosis or it chooses a combination of proposals that minimizes the number of reconfigurations and impact in the smart grid. If it does not receive any information or proposals within a user configurable time period or if they are not sufficient, a warning is posted in the user interface. In recovery, it accepts the proposals that contribute to restoring the smart grid to a planned configuration or improve its state and,

consequently, the states of the supported processes. If the acceptance is acknowledged, requiring again a sufficient number of consistent messages, it proceeds to the reconfiguration. If it is denied, it resumes to a following self-healing cycle, repeating the negotiation.

The cooperation activity halts or reserves requested resources indefinitely.

The cooperation module executes in parallel with the remaining activities. As a SHEE receives cooperation requests from other SHEEs, it stores each in memory until a sufficient number of consistent messages have been received from the corresponding replicas. If new requests from the same SHEE are received, requiring again a sufficient number of consistent messages, the previous are discarded. It processes and replies to the requests as the number of received consistent messages complies with the previous requirement. The reply comprises a reconfiguration proposal to which a set of resources might be temporarily allocated in the knowledge model. This reconfiguration will only be applied upon its acceptance by the other SHEE, requiring again a sufficient number of consistent messages. If it is not accepted after a user configurable time period, the resources are released. If the acceptance is received after this period and the resources are still available, then the knowledge model is updated with the necessary information to trigger the self-healing process that will provide the required reconfiguration. The requester is notified of this action. If not, the reconfiguration is denied, requiring the requester to restart the negotiation.

4.5.2 Safety Validation

The safety validation focus on the contents of the knowledge model, the coordination between replicas and the command execution in the smart grid systems and components.

A cooperating SHEE or the smart grid systems and components execute replayed, outdated and/or out-of-sequence requests or commands.

The employment of secure protocols prevents this scenario from happening at communications level. At application level, the issued commands and cooperation requests are marked with a time stamp, a reconfiguration plan identification and a sequence number. If the message is older than a user configurable time period, it is discarded. If it corresponds to a new reconfiguration plan, the messages corresponding to a previous plan are disregarded. If it arrives out of sequence, it will be stored in memory until the previous messages have arrived. Moreover, each request or command is only processed after a sufficient number of consistent messages from the replicas are received.

The knowledge is outdated.

As we have previously explained, the collectors collect smart grid information continuously, as long as there is activity from the smart grid information sources and/or scheduled events. Only timely information is processed and in the right sequence. Moreover, the SMR module executes continuously, as long as $n - f$ replicas are available. The information batches delivered from the SMR to the knowledge module are updated to the knowledge model. Therefore, if there are smart grid events, they will be reflected in the knowledge model. If less than $n - f$ replicas are available, the knowledge model is not updated. In this scenario, there being no updates means that the monitoring activity is not triggered. Therefore, it does not work with the outdated knowledge.

The self-healing activities in different replicas consider different smart grid information in the same execution cycle.

All the collected smart grid information that is relevant to the knowledge module is submitted to the SMR module. By using a SMR protocol, the latter coordinates between the replicas which information will be submitted to the knowledge module in each batch and by which order. The knowledge module at each replica eventually updates the knowledge model, creates a copy of it and triggers the monitoring activity. All the activities work with this copy during the current self-healing cycle. Therefore, the self-healing activities in different replicas consider the same knowledge in the same execution cycle.

The different replicas decide different reconfiguration plans.

The knowledge model at different replicas contains the same facts and rules in the same order for the same self-healing cycle. Likewise, the remaining modules at different replicas contain the same goals and queries. Moreover, we have already shown that the self-healing activities in different replicas consider the same knowledge in the same execution cycle. Therefore, the monitoring, diagnosis and recovery provide the same results in different replicas, which, consequently, decide the same reconfiguration plan.

A reconfiguration proceeds disregarding the consequences of its middle steps.

During a reconfiguration, the corresponding module compares the resulting smart grid reconfiguration and consequences with what it expects from issuing a command at each step. If there is a divergence between both, the current reconfiguration is interrupted and the self-healing process is resumed to a next cycle. Moreover, during a reconfiguration, new self-healing cycles are triggered by the knowledge updates. In each cycle, the monitoring, diagnosis and recovery activities create a new recovery plan which is compared with the ongoing reconfiguration. If it is consistent, the new plan is discarded. If it is inconsistent, the ongoing reconfiguration is interrupted and a new reconfiguration is started,

based on the new recovery plan.

The SHEE modules work with inconsistent knowledge.

The monitor validates the knowledge consistency through the comparison of redundant knowledge. If an inconsistency is detected, a warning is posted in the user interface and the inconsistent information is disregarded by the self-healing activities. To allow this comparison, the deployment should enable the SHS connectivity with redundant smart grid information sources.

The cooperating SHEE or the smart grid systems and components execute illegitimately altered requests and/or commands.

By employing secure communication protocols, the SHEE and the smart grid systems and components validate the integrity of the received messages. Moreover, each request or command is only processed after a sufficient number of messages from the replicas are received, which are compared with each other to validate their consistency. In this context, if a request or command is considered invalid, it is disregarded.

The cooperating SHEE or the smart grid systems and components execute requests and commands from anyone, including faulty or malicious replicas.

A SHEE knows the SHEEs in contiguous self-healing domains, from which it might receive cooperation requests. Likewise, the smart grid systems and components know the SHEE corresponding to their self-healing domain. When establishing communications and by using secure protocols, both authenticate the other party. Moreover, each request or command is only processed after a sufficient number of consistent messages from the replicas are received. Therefore, only known, authenticated and replica consistent requests and commands are processed.

4.6 Discussion

There are design and implementation issues that depend on each specific smart grid context, namely, the deployed infrastructure, specific risks, company's policies, financial aspects and project deadlines, some of which have already been mentioned in Chapter 3. To deal with these issues, we created a flexible and agile SHS that achieves these properties through its modularity and knowledge-based reasoning. In this regard, different SHEE collectors and controllers can be added to communicate with different information sources and controlled devices. The knowledge rules, goals and queries can be updated to reflect the behavior of these components and the supervision of new services. The level of distribution and replication can also be adapted, depending on specific needs. In

the following sections, we raise what we consider to be the most relevant issues when implementing the SHEE, providing guidance for future deployments.

4.6.1 Creating a Self-healing Ecosystem

A smart grid might already have other electrical network self-healing and cyber security automation solutions deployed. Depending on what the purpose and behavior of each specific solution, the DSO or manufacturer must decide to separate the scope of these solutions from the scope of the SHS or to integrate both. If the choice is to separate the scopes, then the smart grid systems and components of the deployed solutions must be excluded from any SHEE self-healing domain, to avoid concurrency between the automatic actions of both. If the choice is to integrate both, then the necessary collectors and controllers must be developed and/or integrated in the SHEEs that will have the aforementioned solutions in their self-healing domain. The knowledge rules, goals and queries must be adapted to the corresponding facts so that the decisions and actions of the SHS are coordinated with the actions and state of these solutions.

4.6.2 Using the Existing Infrastructure in the SHS

The SHS can take advantage of an already deployed SCADA infrastructure to have access to the smart grid components, information and control capabilities, both at the local systems and central systems levels, avoiding the deployment of a dedicated self-healing infrastructure and despite creating an additional dependence. For example, in a smart grid as a whole self-healing domain, a SHEE hosted at a data center could take advantage of the neighbor SCADA server and of the network, hosting and security monitors (e.g., a SIEM or other monitoring platform) to have complete visibility over the smart grid. The communication protocols required by each SHEE depend on deciding how it will collect each necessary piece of information and send each reconfiguration command - by itself versus taking advantage of the deployed information repositories and smart grid controllers - in the context of its self-healing domain. SHEE collectors and controllers must be developed to support the necessary protocols. A rough comparison between using a dedicated self-healing communications infrastructure and using a shared communications infrastructure is presented in Table 4.1.

SHEE Support for Control and Data Acquisition

The monitoring and control of the sensors and actuators and of the SCADA control systems and components commonly requires the support for standard-based protocols at the field level and for proprietary protocols at the data center level, which have been identified in the Section 2.4.9. For the field level, there are currently two main open source implementations of IEC 61850 and IEC 60870-5-104 communication protocols: libIEC61850

[82], in Java, and OpenMUC [83], in standard C, which can be used as basis for the SHEE implementation.

Table 4.1: Self-healing communications infrastructure deployment comparison.

Type	Advantages	Disadvantages
Dedicated	It can be designed for the specific self-healing use case; It is only exposed to its own risks.	It is necessary to deploy a new communications infrastructure; It may be necessary to implement proprietary communication libraries; It is necessary to implement the necessary safety procedures.
Shared	The communications infrastructure is already deployed; Some of the required data has already been centralized by SCADA, health, performance and security monitoring components.	Possibly, the existing communications infrastructure needs to be reinforced; It may be necessary to implement proprietary communication libraries.

The IEC 62351 series (namely, the IEC 62351-3:2014 and IEC TS 62351-4:2007) and the IEC TS 60870-5-7:2013 define the security extensions for IEC 61850 and IEC 60870-5-104, respectively. Regarding the data center level, as most SCADA servers implementations are proprietary, the manufacturers must be involved in the SHEE implementation process. Even if the monitoring is accomplished through the direct connection to the servers' databases with a read-only profile, while complying to the in-place security controls and without going through the application, the manufacturer must still provide information about the data base structure and security.

Network and Security

The network and security systems and components commonly use standard-based protocols for logging and management, as it was explained in the Section 2.7. In this regard, the SHEE must implement syslog as a log collector, to gather the logs of the systems and components which cannot send them, and as a log receiver, to get the logs of the systems and components which have the capability to send them and that are configured to do so. The domain systems and components must log all the policy related security events. The SHEE must also implement the SNMP protocol for event receiving and systems and components configuration. The domain systems and components must be configured to send all the policy related health and performance events. If system monitors or log collectors are already in-place and if they are already concentrating and/or collecting the necessary logs and events from and within the SHEE domain, they can be configured to forward that

data to the SHEE. In the latter case, the SHEE could be implemented as only a receiver for monitoring. It may also be necessary to implement also the SSH and HTTPS protocols for the systems and components configuration, depending on which protocol they support. An Internet search for the aforementioned protocols shows that there are a few open source implementations available, such as: Syslog4j [84], in Java, SNMMP4J [85], in Java, and JSch [86], also in Java, which claim to be compliant with the corresponding standards. However, they require further security and robustness testing

4.6.3 Adapting Existing Systems to Create a SHEE

As the SHEE is modular, it can be implemented from scratch or it can be implemented through the adaption of existing commercial systems. The most suitable systems to adapt would be those that have a similar purpose and that already implement partially or completely some of its modules.

Example 1 An example would be to adapt the intelligent self-healing approaches of Section 2.6, which have the necessary electrical network monitoring and control capabilities, either centralized or distributed, for a given set of secondary substations. Regarding the centralized approach, it is similar to a SHEE that has the electrical network layer of a set of secondary substations as its self healing domain. In the case of the distributed approach, it is similar to a set of SHEEs that have as self-healing domain the electrical network layer of the corresponding secondary substations. As these solutions are restricted to a set of components, to the electrical network layer and to a predetermined group of situations, their adaption would require: support for the easy update and detailed characterization of the monitored components, the integration of collectors and controllers for other smart grid layers' components, the evolution to an agile self-healing engine that is able to create plans based only on the available smart grid information and on the results from past actions - predefined conditions should not be used -, the ability to cooperate with other instances and the assessment of their security and dependability implementation.

Example 2 Another example would be to adapt a SIEM system, which has the necessary capabilities to monitor the security of a communications network. Although it does not control the monitored components, it can trigger the forwarding of report messages and alarm messages. It is similar to a SHEE that has the communications network layer of the smart grid as its self-healing domain. Unlike the previous example, it can have information from all smart grid layers, if the necessary collectors are available, and it is not able to control any smart grid components. Like the previous example, it is restricted to a set of predetermined situations (that are configured by the users as needed). Its adaption would require: support for the detailed characterization of the smart grid components, the development of controllers for

all smart grid layers, the eventual development of further collectors, the definition of generic rules that are based only on the available smart grid information and on the results from past actions - once again, predefined conditions should not be used -, the ability to cooperate with other instances and the assessment of their security and dependability implementation.

4.6.4 Restrictions and Limitations to the Self-healing Process

The decisions and actions taken by the SHS are limited by the available resources during a self-healing cycle (e.g., not enough knowledge, none to reduced control capability and/or no connectivity), which means that it is not a certainty that it will be able to go through all of its states and, consequently, heal the smart grid in every cycle. However, the monitoring and diagnosis states will always produce relevant information which can be used by the human operators to complete the healing process. In regard to the knowledge restrictions, the knowledge model is a limited representation of reality, which is more accurate as closer it is to the real world. Making the correct decisions and taking the right actions depends on having the necessary information on time, representing it as knowledge in the right way and making the right questions. A harmful plan might result from insufficient, false (as acquired from the smart grid components), outdated and/or wrong knowledge and not by system malfunctioning, as long as $f + 1$ replicas are available and executing correctly. For the same reasons, the SHS may try to prevent or reverse operator's legitimate configurations that are not properly reflected in the SHEEs' knowledge models, considering them as faults. The same issue can affect the human experts and the way to address it is through strengthening and adapting one's capabilities, teaching, training and self-learning. The same approach can be used with the SHS, through the use of faster and more robust communication channels, improving the self-healing ontology and the knowledge representation based on experience and providing the learning module with improved self-learning capabilities, beyond classifying the effectiveness of the issued reconfiguration commands. Nevertheless, there might be situations in which the proposal-only or the priority operator's actions modes must be used during short time periods.

4.6.5 Distributing the SHS and Replicating the SHEEs

A highly distributed SHS distribution level and the use of more than four replicas per SHEE increase the deployment, management, operation and maintenance costs of the proposed self-healing solution. In regard to the distribution, the Table 4.2 compares the number of deployed SHEEs in a set of possible scenarios, which range from a small set in a centralized redundant scenario to the tens of thousands in a highly granular scenario. Given the threat of communications network partitioning, a DSO must evaluate the dif-

ferent risk levels associated with the various network locations and use a more granular SHEE distribution in locations with higher risk (i.e., with lower communications redundancy and/or higher criticality), thereby reducing the associated costs.

Table 4.2: Distributed deployment comparison.

Type of distribution	Numbers
Centralized (1)	$1 \times \text{SHEE} = \text{One}$
By data center (n_{dc})	$n_{dc} \times \text{SHEE} \approx \text{A small set}$
By primary substation ($n_{ps} > 100$)	$(n_{dc} + n_{ps}) \times \text{SHEE} \approx \text{A couple of hundreds}$
By secondary substation ($n_{ss} > 10k$)	$(n_{dc} + n_{ps} + n_{ss}) \times \text{SHEE} \approx \text{Tens of thousands}$

The BFT SMR increases the distribution numbers, depending on the number of replicas per SHEE. The minimum number of replicas is four, to tolerate one fault or intrusion in one of the replicas. However, depending on the risks to which a SHEE is exposed, a higher number of replicas might be used. The DSO may reduce the deployment costs by optimizing the resource utilization. In this regard, the Table 4.3 compares a set of replica deployment scenarios, starting with a single machine to the case where all replicas are distributed by different facilities.

Table 4.3: Replication deployment comparison (n is the number of SHEE replicas).

Type	Deployment resources	Unique points of failure
Different threads in the same machine	$n \times \text{SHEE application}$ + $1 \times \text{OS}$ + $1 \times \text{Computer hardware}$ + $1 \times \text{Facility}$	OS; Computer hardware; LAN; Facility
Different VMs in the same host	$n \times \text{SHEE application}$ + $n \times \text{OS}$ + $1 \times \text{Hypervisor}$ + $1 \times \text{Computer hardware}$ + $1 \times \text{Facility}$	Hypervisor; Computer hardware; LAN; Facility
Different local hosts	$n \times \text{SHEE application}$ + $n \times \text{OS}$ + $n \times \text{Computer hardware}$ + $1 \times \text{LAN}$ + $1 \times \text{Facility}$	LAN; Facility
Different geographically distributed hosts	$n \times \text{SHEE application}$ + $n \times \text{OS}$ + $n \times \text{Computer hardware}$ + $n \times \text{LAN}$ + $n \times \text{Facility}$ + $1 \times \text{WAN}$	WAN

Each scenario creates different unique points of failure, which must also be assessed for the corresponding risks in the specific context of each SHEE.

A proper SHS distribution granularity, together with a sufficient number of adequately distributed replicas per SHEE, enable the SHS to avoid communications network partitioning scenarios that can range in size from a single sensor to a set of facilities/locations. Figure 4.8 depicts a set of substations and DA devices with the expected connections between one-another. There is a replicated SHEE in each primary and secondary substation. The SHEEs at primary substations 1 and 2 supervise not only the primary substation but also the MV electrical network and corresponding DA devices. The green lines represent the expected connections between the SHEEs having service related self-healing domains. Figure 4.9 depicts the same set of substations and DA devices in four different partitioning example scenarios, in comparison with Figure 4.8, affecting a control center, a substation, a substation group or a data center. At each scenario, the SHEEs maintain supervision over the corresponding self-healing domain components that stay in their partition, being capable to prevent and recover from failures by healing the faults affecting these components. In this regard, if a partition is caused by faulty configurations, the SHEEs at each side might be capable of removing the fault, by correcting the configurations, restoring the smart grid connectivity.

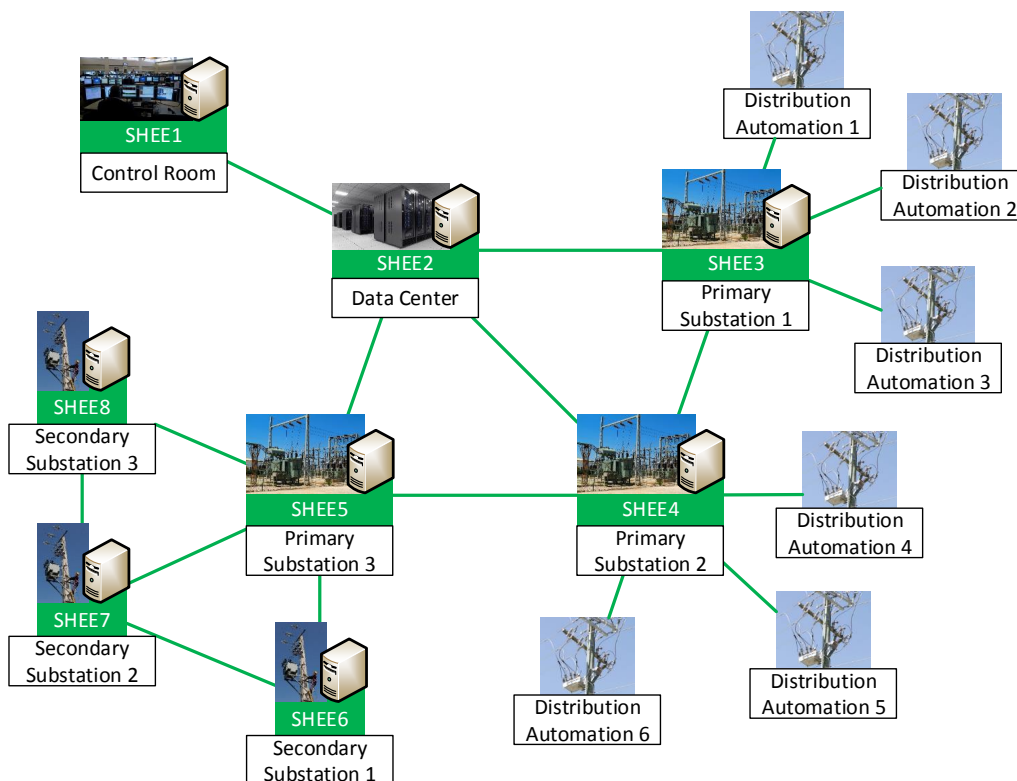
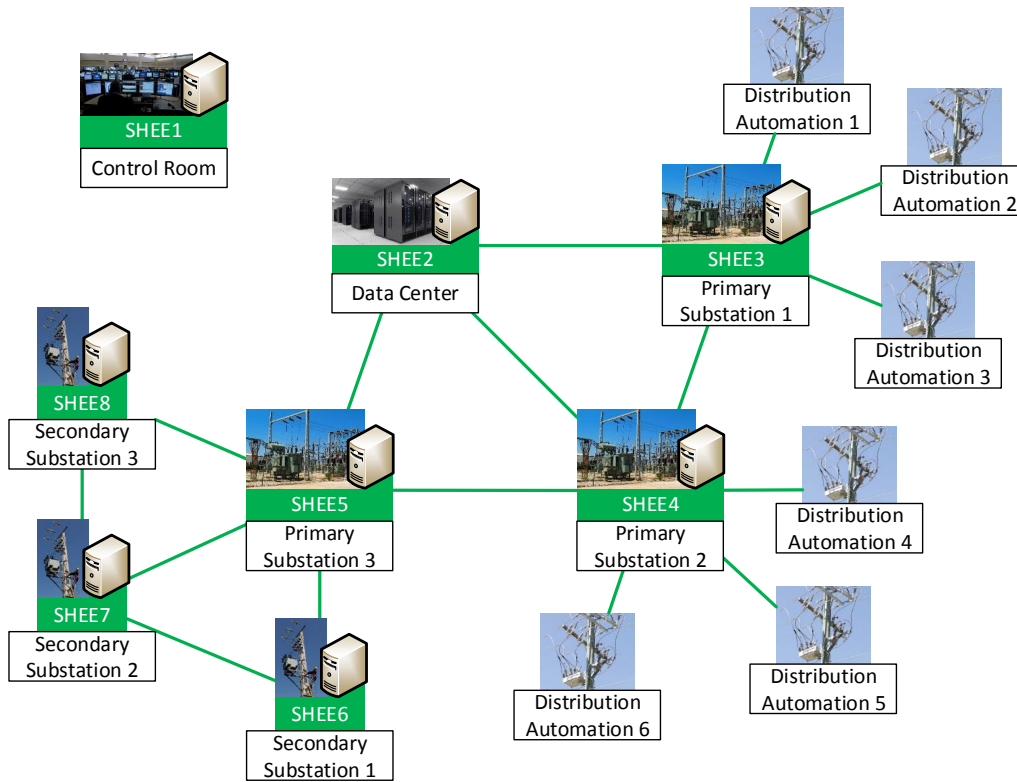


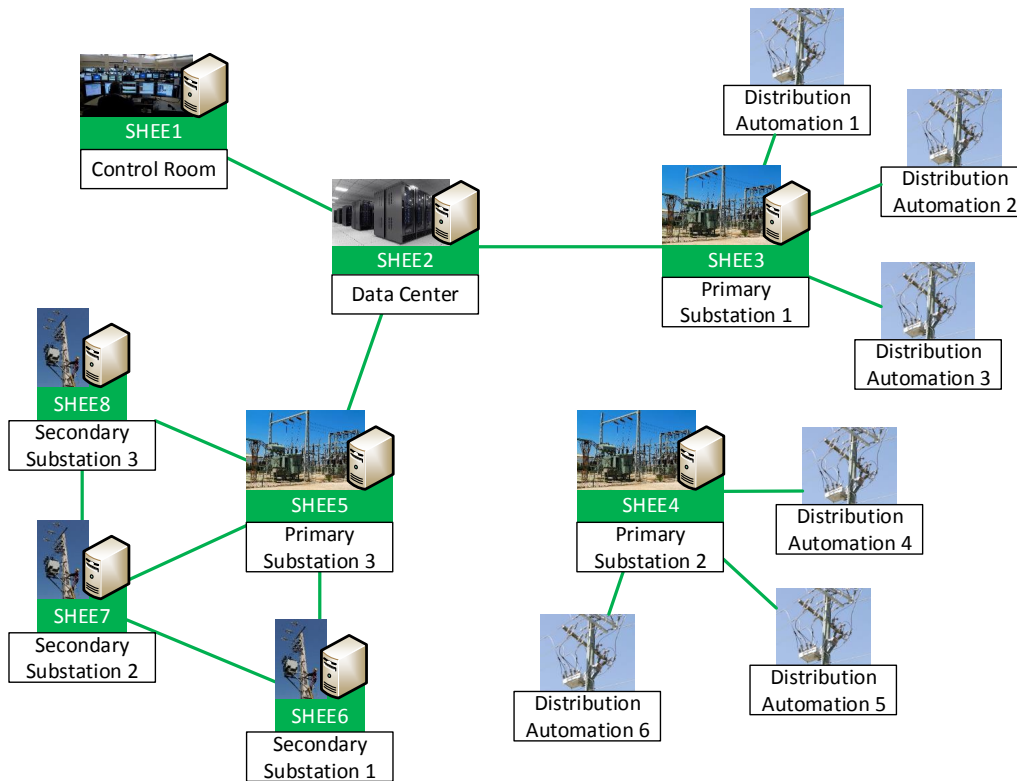
Figure 4.8: Self-healing system without network partitions.

4.6.6 Closing Remarks

Considering that the design assumptions hold and that the DSO and manufacturers follow the provided guidelines through the implementation, deployment, management, operation and maintenance stages of the SHS life cycle, the SHS will behave in compliance with the design requirements, providing a valid solution to the design problem. Moreover, moving beyond the automatic smart grid self-healing context, the SHS provides information regarding the cause and impact of failures, considering all the smart grid technical layers, that might be used as guidance by incident response teams to reduce response and recovery times and by management teams to prevent incident recurrences.

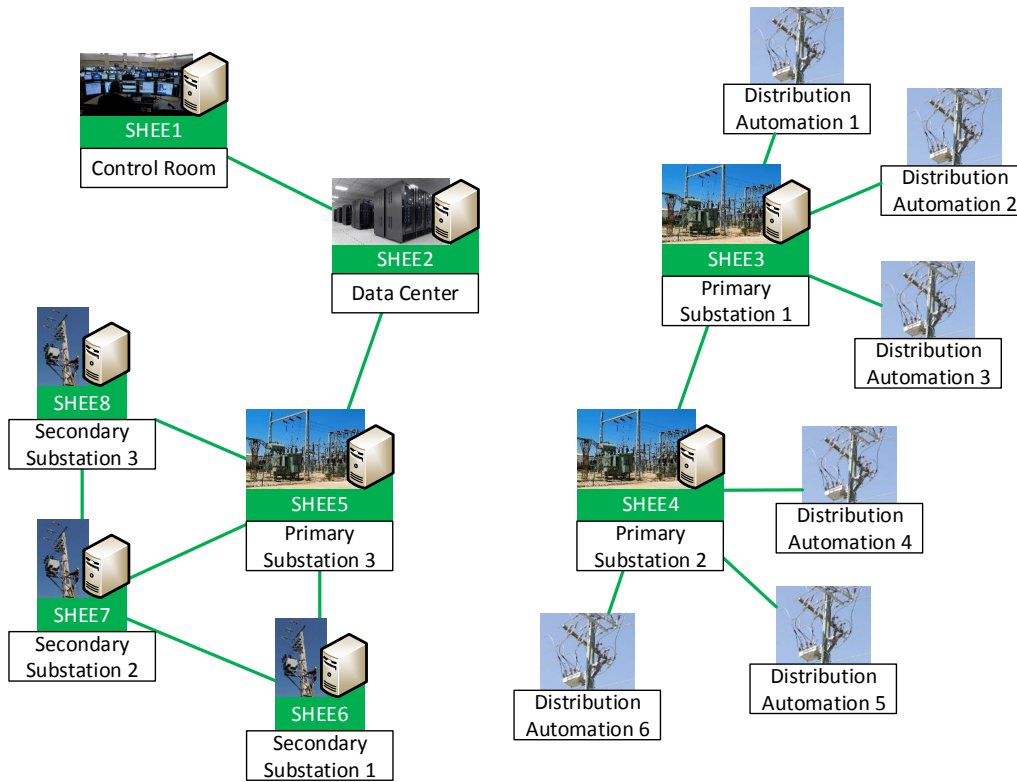


(a) An isolated control center.

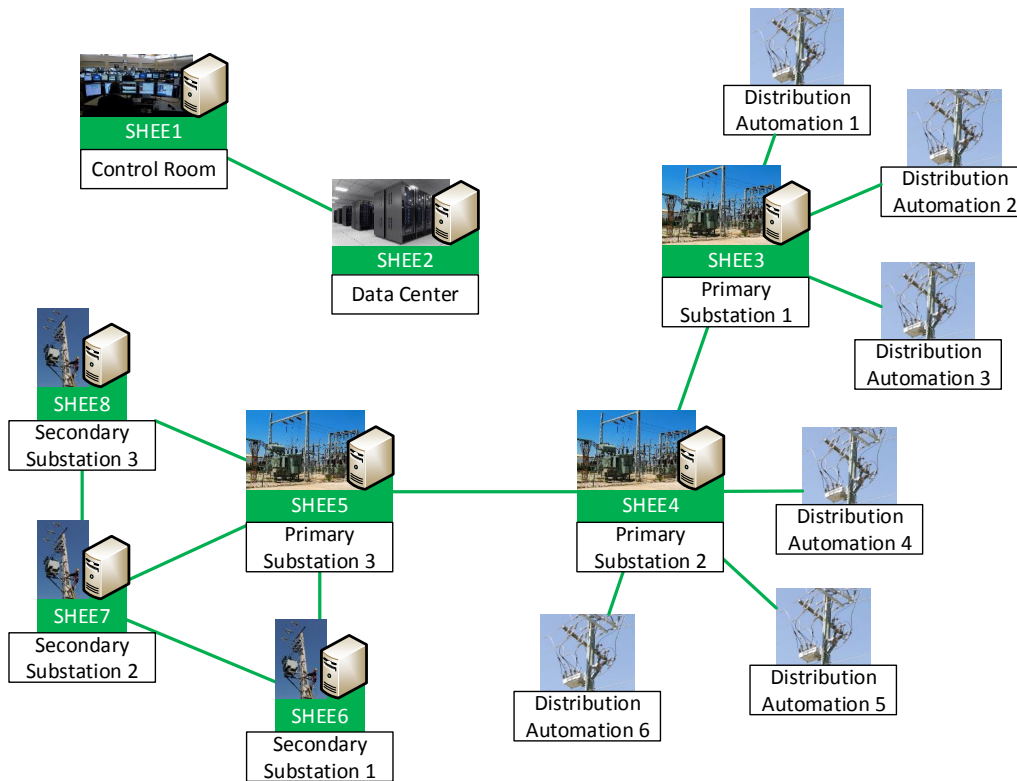


(b) An isolated substation.

Figure 4.9: Network partition examples.



(c) An isolated substation group.



(d) An isolated data center.

Figure 4.9: Network partition examples.

Chapter 5

Conclusion

This chapter concludes the current work by presenting a summary of the results and listing a set of remaining issues to be handled in future work.

5.1 Results

Performance, privacy, security and dependability are key smart grid concerns, which are related with the handling of real-time data and sensitive information and with the proven fact that it is exposed and vulnerable to different threats, ranging from the traditional electrical faults to the more recent cyber-attacks. In the highly connected smart grid environment, the manual interventions, faults, self-healing configurations or intrusion prevention actions, observed in one of its layers, may have hindering or harmful consequences in the systems, components and processes of the other layers. These consequences can only be timely foreseen, avoided or unveiled by correlating the right knowledge about the smart grid systems, components, processes, their behavior and their connections with the real-time electrical, health, performance and security events. This is the role of the operational planning teams and incident response teams with the appropriate HR capacity, training and technological support. However, it is uncommon for the afore mentioned systems to natively support or be configured to share the required information among them. Moreover, a smart grid incident response capability that over-relies on the central systems is vulnerable to non-malicious faults or successful cyber attacks that can partially or completely isolate the latter from the field networks, blocking the view over what is happening in one or more smart grid layers and forcing the dispatch of teams to the affected areas. The field devices must rely on the local security features, until the field teams arrive or the communications are reestablished.

We propose a secure and dependable distributed expert system as a way to provide a more efficient, secure and dependable self-healing capability to the electricity distribution smart grid. We modeled the smart grid electrical network, communications network, SCADA systems and processes to understand how they can be described as facts and rules

of an expert system knowledge model. We perform a threat and vulnerability assessment on the self-healing process to identify the required security and dependability controls, focusing on the threats to the self-healing process and on the threat that a faulty or malicious self-healing capability might mean to the smart grid. As a result of the assessment, we present a system architecture that is comprised of a given number of actively replicated SHEEs, with assigned self-healing domains, that can be geographically distributed through the smart grid. Each SHEE has a knowledge model with facts and rules that describe the smart grid systems, components and processes, including their behavior and connections, that fall within its self-healing domain. The SHEE can connect to them to collect the real-time electrical, health, performance and security events that are used to update and reason with the knowledge model and to control and/or reconfigure them. Reasoning gives to the SHEE the intelligence needed to diagnose failures and create recovery plans. It will ask for the cooperation of neighbor SHEEs when the information or resources within its domain are not sufficient to complete the self-healing process. To reduce its vulnerability to network partitions, a SHEE must be as connectively close as possible to its self-healing domain. A SHEE also provides support for user interaction and user, configuration, logging, backup and cryptography management.

We implemented a simple POC, with focus in the knowledge modeling and reasoning, to demonstrate how to describe the smart grid systems and components as facts and rules in a SHEE knowledge model. We also demonstrate what goals and queries may be asked to the knowledge model to diagnose smart grid failures and create recovery plans. The results are simple yet promising, validating the underlying concept and setting the foundations for further R&D. We provide guidance for the implementation of the remaining modules, which we were unable to implement in the POC due to time constraints, to test the performance, robustness and security of the complete implementation. We also discuss a set of relevant implementation and deployment aspects, namely the possibility to take advantage of an already existing communications network, data concentrators and SCADA controllers, the number of self-healing domains with a dedicated SHEE and the number of replicas in a SHEE. The main conclusion is that, although we can play with these parameters to change the setup, management, operation and maintenance costs of the proposed solution, this will have consequences in the performance, security and dependability of the latter. Moreover, the creation of the recovery plans will be always restricted by the existing visibility, control and redundancy limitations of the smart grid systems and components.

5.2 Future Work

The following issues must be handled in future work.

5.2.1 Design

The following issues are specific to the SHS design:

- Intelligent supervision components:
 - Evolution of the learning module to include pattern recognition;
 - Development of a prognosis module that uses the learned patterns to anticipate faults and consequent failures;
 - Development of a preventive maintenance module to generate the maintenance plans;
- Expert components:
 - Development of a database to store the collected smart grid information and knowledge model snapshots;
- Smart grid modeling:
 - Model further systems, components and services belonging to the presented smart grid layers;
 - Model the HR management systems, focusing on the smart grid users;
 - Model the service order scheduling systems, focusing on the smart grid interventions;
- Threat and vulnerability assessment:
 - Perform new assessment cycles, updating the list of threats and vulnerabilities to reflect the smart grid context evolution;
 - Perform new assessments from the perspective of other SHS beyond the self-healing application.

5.2.2 Implementation

The following issues are specific to the SHS implementation:

- Knowledge:
 - Formally define and maintain an updated ontology to include the new systems, components and services;
 - Update the facts in compliance with the updated ontology;
 - Improve the reasoning rules to reflect the new facts and to narrow the gap between the modeled behaviors and the real world;

-
- Software code development for each SHEE module;
 - Validation:
 - Validate the security and robustness of the developed code;
 - Discussion:
 - Extend the distribution and replication guidelines with a set of example deployment scenarios.

Acronyms

- AAA** Authentication, Authorization and Accounting. 33
- ACL** Agent Communication Language. 63
- AMI** Advanced Metering Infrastructure. 3, 10, 20, 21
- AMR** Advanced Meter Readings. 20
- BDI** Belief-Desire-Intention. 62
- BFT** Byzantine Fault-Tolerant. 4, 40, 64, 65, 70, 72, 75, 76, 98, 106
- CCTV** Closed-Circuit Television. 32
- CDH** Central Data Hub. 16
- CFP** Call for Proposals. 63
- COSEM** Companion Specification for Energy Metering. 21
- COTS** Commercial off-the-shelf. 31
- CU** Central Unit. 19
- DA** Distribution Automation. 2–4, 10, 17, 20, 25, 48, 53, 107
- DAM** Data Access-Point Manager. 16
- DC** Data Concentrator. 20
- DER** Distributed Energy Resource. 30, 36
- DG** Distributed Generation. 12–14, 17, 21
- DLMS** Device Language Message Specification. 21, 29, 30
- DLP** Data Loss Prevention. 61, 131
- DMS** Distribution Management System. 18, 19, 26, 27, 38, 47, 53, 64

- DPIA** Data Protection Impact Assessment. 23, 35
- DR** Demand Response. 14
- DSM** Demand Side Management. 14
- DSO** Distribution System Operator. 9, 12, 13, 15–21, 23, 24, 28, 33, 35, 36, 39, 41, 48, 53, 54, 56, 59, 64, 65, 90, 102, 105, 106, 108
- DTC** Distribution Transformer Controller. 20
- EC** European Commission. 13, 14, 16, 21, 36
- EG2** Expert Group 2. 23
- EG3** Expert Group 3. 15
- EOL** End-of-life. 35
- EU** European Union. 13, 14
- EV** Electric Vehicle. 13, 15
- FDLIR** Fault Detection, Location, Isolation and Restoration. 66
- FIPA** Foundation for Intelligent Physical Agents. 63
- FLIR** Fault Location, Isolation and Restoration. 3, 10, 25, 28, 36, 64
- FPI** Fault Passage Indicator. 28
- GDPR** General Data Protection Regulation. 23
- GIS** Geographic Information System. 18, 19, 23, 26
- GPRS** General Packet Radio Service. 21
- GPS** Global Positioning System. 19
- GSM** Global System for Mobile Communications. 21
- GUI** Graphical User Interface. 19
- HAN** Home Area Network. 20
- HIM** Historical Information Manager. 18
- HMI** Human-Machine Interface. 19, 20, 53, 57, 81

- HR** Human Resources. 31, 60, 111, 113, 130, 131
- HTTPS** Hyper Text Transfer Protocol Secure. 33, 104
- HV** High Voltage. 12, 17, 26, 48
- I/O** input/output. 4, 32, 40, 59, 64
- ICS** Industrial Control Systems. 21, 22, 24, 31, 33, 34
- ICT** Information and Communication Technologies. 31, 36
- IDS** Intrusion Detection System. 3, 10, 33, 35
- IED** Intelligent Electronic Device. 19, 47, 51, 53, 74, 81, 83
- IP** Internet Protocol. 21, 51
- IPS** Intrusion Prevention System. 33
- ISMS** Information Security Management System. 35, 59
- JRC** Joint Research Centre. 15, 35
- KIF** Knowledge Interchange Format. 63
- KQML** Knowledge Query Manipulation Language. 63
- LAN** Local Area Network. 19, 21, 106
- LV** Low Voltage. 12, 17, 29, 30, 48, 53
- MAS** Multi Agent System. vii, 4, 40, 61–63, 65
- MG** Micro-Generation. 2, 9, 12, 14, 17, 30, 54
- MPLS** Multiprotocol Label Switching. 21
- MV** Medium Voltage. 4, 12, 17, 20, 25–31, 48, 51, 53, 64, 74, 107
- NMS** Network Management System. 34
- NTP** Network Time Protocol. 19
- OMS** Outage Management System. 18
- Operation Technology** Technology that is dedicated to the operation of an infrastructure, such as a SCADA system. 31, 34

- OS** Operating System. 32, 35, 58, 60, 65, 106, 127, 129–131
- OWL** Web Ontology Language. 46
- PAM** Privileged Access Management. 33
- PDH** Plesiochronous Digital Hierarchy. 21
- PKI** Public Key Infrastructure. 33
- PLC** Power Line Communication. 21, 29, 51
- POC** Proof of Concept. 5–7, 79, 80, 112, 133
- prosumer** An electricity consumer that is also a producer. 16, 17
- PV** Photovoltaic. 30
- QoS** Quality of Service. 2, 10, 13
- R&D** Research and Development. 3, 9, 10, 16, 25, 31, 35, 36
- RBAC** Role-based access control. 32, 59, 128
- RDF** Resource Description Framework. 46
- RES** Renewable Energy Source. 11–14, 36
- RTU** Remote Terminal Unit. 20, 28, 29, 47, 51, 53, 64
- SC** Substation Controller. 19, 20, 38, 47, 53
- SCADA** Supervisory Control and Data Acquisition. 2, 10, 13, 18–20, 23, 24, 26, 28, 37, 38, 47, 51, 53–56, 64, 66, 70, 73, 74, 83, 87, 88, 91, 102, 103, 111, 112
- SDH** Synchronous Digital Hierarchy. 21
- SGAM** Smart Grid Architecture Model. 46
- SGCG** Smart Grid Coordination Group. 46
- SGTF** Smart Grid Task Force. 15, 23
- SHA** Self-Healing Application. 57–60, 64
- SHE** Somewhat Homomorphic Encryption. 36, 101

-
- SHEE** Self-Healing Expert Entity. vii, 4–7, 39, 40, 65, 66, 70–77, 79–81, 85, 87, 88, 90, 97–99, 101–107, 112, 114
- SHS** Self-Healing System. vii, 2–4, 6, 7, 39–44, 47, 57–60, 64–66, 79–81, 97, 98, 101, 102, 105, 107, 108, 113
- SIEM** Security Information and Event Management. 3, 10, 34, 47, 102, 104
- SMR** State Machine Replication. 4, 40, 64, 65, 70, 72, 75, 76, 98, 100, 106
- SNMP** Simple Network Management Protocol. 34, 103
- SSC** Smart Substation Controller. 27
- SSH** Secure Shell. 33, 104
- TSO** Transmission System Operator. 12, 21
- URI** Uniform Resource Identifier. 46, 47
- VHV** Very High Voltage. 12
- VLAN** Virtual Local Area Network. 32, 51, 54, 61, 130
- VM** Virtual Machine. 32, 60, 64, 106, 127
- VPN** Virtual Private Network. 33, 61, 130
- VRF** Virtual Routing and Forwarding. 32, 61, 130
- W3C** World Wide Web Consortium. 46
- WAN** Wide Area Network. 19, 74, 106

Bibliography

- [1] G. Veronese, M. Correia, A. Bessani, L. Lung, and P. Verissimo. Efficient byzantine fault-tolerance. *In IEEE Transactions on Computers*, 2013.
- [2] Entidade Reguladora dos Serviços Energéticos (ERSE). Electricity activities, Available in <http://www.erse.pt/eng/electricity/Activities/Paginas/default.aspx>, Accessed in December 2015.
- [3] BP. BP statistical review of world energy, June 2015.
- [4] Eurostat, European Commission (EC). Electricity production and supply statistics, Available in http://ec.europa.eu/eurostat/statistics-explained/index.php/Electricity_production_and_supply_statistics, Accessed in January 2015.
- [5] European Commission (EC). Priorities, Available in http://ec.europa.eu/priorities/index_en.htm, Accessed in January 2016.
- [6] European Commission (EC). Energy union and climate, Available in http://ec.europa.eu/priorities/energy-union/index_en.htm, Accessed in January 2016.
- [7] European Commission (EC). Energy strategy, Available in <https://ec.europa.eu/energy/en/topics/energy-strategy>, Accessed in January 2016.
- [8] European Commission (EC). Energy 2020: A strategy for competitive, sustainable and secure energy, 2010.
- [9] European Commission (EC). Energy roadmap 2050, 2011.
- [10] The European Parliament and The Council of the European Union. Directive 2009/28/EC on the promotion of the use of energy from renewable sources, 2009.
- [11] The European Parliament and The Council of the European Union. Directive 2014/94/EU on the deployment of alternative fuels infrastructure, 2014.
- [12] C. Thiel, J. Krause, and P. Dilara. Electric vehicles in the EU from 2010 to 2014 - Is full scale commercialisation near?, 2015.

- [13] Smart Grid Task Force. EG3 first year report: Options on handling smart grids data, 2014.
- [14] DG Energy, European Commission (EC). Smart grid mandate: Standardization mandate to European Standardisation Organisations (ESOs) to support European smart grid deployment, 2011.
- [15] IEC. Iec 60947-2:2016, low-voltage switchgear and controlgear - part 2: Circuit-breakers, 2016.
- [16] Energias de Portugal (EDP). O que provoca as interrupções de energia, Available in <https://www.edp.pt/pt/negocios/interrupcoesenergia/Pages/OqueProvocaAsInterrupcoesDeEnergia.aspx>, Accessed in May 2016.
- [17] ICS-CERT. Trends in incident response in 2013, 2013.
- [18] ICS-CERT. Incident response/vulnerability coordination in 2014, 2015.
- [19] European Union Agency for Network and Information Security (ENISA). Eu agency analysis of "stuxnet" malware: a paradigm shift in threats and critical information infrastructure protection, Available in <https://www.enisa.europa.eu/media/press-releases/eu-agency-analysis-of-2018stuxnet2019-malware-a-paradigm-shift-in-threats-and-critical-information-infrastructure-protection-1>, Accessed in December 2010.
- [20] Symantec. Targeted attacks against the energy sector, 2014.
- [21] SANS ICS and E-ISAC. Analysis of the cyber attack on the Ukrainian power grid: Defense use case, 2016.
- [22] The European Parliament and The Council of the European Union. General data protection regulation, 2016.
- [23] Article 29 Data Protection Working Party. Opinion 12/2011 on smart metering, 2011.
- [24] Expert Group 2: Regulatory Recommendations for Privacy, Data Protection and Cyber-Security in the Smart Grid Environment, Smart Grid Task Force 2012-2014, European Commission. Data protection impact assessment template for smart grid and smart metering systems, 2014.
- [25] Autoridade Nacional da Protecção Civil (ANPC). Infraestruturas críticas, Available in <http://www.prociv.pt/RISCVULNERABILIDADES/Pages/Infraestruturas-Criticas.aspx>, Accessed in December 2015.

- [26] U.S. Department of Homeland Security (DHS). What is critical infrastructure, Available in <http://www.dhs.gov/what-critical-infrastructure>, Accessed in December 2015.
- [27] J. Rosa, C. Cândido, F. Ramalheira, P. Marques, R. Fiteiro, A. Leandro, N. Pires, P. Gama, R. Oliveira, and M. Morgado. Last generation reclosers for MV overhead lines at EDP Distribuição - results and conclusions. *In International Conference on Electricity Distribution*, June 2015.
- [28] N. Pereira, M. Louro, A. Leitao, C. Pinto, P. Viegas, and D. Cabral. Integration of fault location in a smart grid operating system. *In International Conference on Electricity Distribution*, June 2013.
- [29] F. Melo, C. Cândido, C. Fortunato, N. Silva, F. Campos, and P. Reis. Distribution automation on LV and MV using distributed intelligence. *In International Conference on Electricity Distribution*, June 2013.
- [30] C. Fortunato, R. Almeida, F. Melo, and C. Cândido. Distribution automation architecture for fault detection and isolation. *In Automation and Control (PAC) World Conference*, July 2015.
- [31] E. Coster, W. Kerstens, and T. Berry. Self healing distribution networks using smart controllers. *In International Conference on Electricity Distribution*, June 2013.
- [32] Schneider-Electric. Stedin: A self-healing utility, Available in <https://www.youtube.com/watch?v=YiaihcpPBG4>, Accessed in January 2016.
- [33] Siemens. Self-healing grid Stedin, Available in <http://w3.siemens.com/smartgrid/global/en/projects/pages/stedin.aspx>, Accessed in February 2016.
- [34] Siemens. Self-healing grid Rotterdam harbor, Available in <https://www.youtube.com/watch?v=72ttBIxz1p4>, Accessed in January 2016.
- [35] SEGRID. D1.1 architecture and design for use cases, April 2015.
- [36] e-balance. D2.1 selection of representative use cases, 2014.
- [37] e-balance. D3.1 high level system architecture specification, 2014.
- [38] e-balance. D3.2 detailed system architecture specification, 2015.
- [39] e-balance. D6.1 specification of the demonstrators, 2015.
- [40] SEGRID. D2.3 segrid gap analysis, April 2016.

- [41] M. Correia and P. Sousa. *Segurança no Software*. 2010.
- [42] W. Stallings and L. Brown. *Computer Security: Principles and Practice*. Second edition edition, 2012.
- [43] P. Veríssimo and L. Rodrigues. *Distributed Systems for System Architects*. 2001.
- [44] C. Cachin, R. Guerraoui, and L. Rodrigues. *Introduction to Reliable and Secure Distributed Programming*. Second edition edition, 2011.
- [45] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Fifth edition edition, 2011.
- [46] B. Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C*. Second edition edition, 1996.
- [47] vmware. Virtualization: How it works, Available in <http://www.vmware.com/uk/solutions/virtualization.html>, Accessed in August 2016.
- [48] Syslog.org. Log management, Available in <http://www.syslog.org/logged/>, Accessed in January 2016.
- [49] SNMP Research International, Inc. The SNMP protocol, Available in <http://www.snmp.com/protocol/>, Accessed in January 2016.
- [50] JRC, European Commission (EC). Smart grid projects outlook 2014, 2014.
- [51] SPARKS. D1.6 overview of research projects in the area of smart grids, March 2015.
- [52] e-balance. Balancing energy production and consumption in energy efficient smart neighbourhoods, Available in <http://www.e-balance-project.eu/>, Accessed in December 2015.
- [53] HEAT. Homomorphic encryption applications and technology, Available in <https://heat-project.eu/>, Accessed in December 2015.
- [54] LV-Pri20. LV-Pri20 project website, Available in http://cordis.europa.eu/project/rcn/196089_en.html, Accessed in December 2015.
- [55] OSGP Alliance. The open smart grid protocol, Available in <http://www.osgp.org/>, Accessed in December 2015.
- [56] SEGRID. Securing the smart grid of tomorrow, Available in <http://www.segrid.eu/>, Accessed in December 2015.

- [57] SPARKS. Smart grid protection against cyber attacks, Available in <https://project-sparks.eu/>, Accessed in December 2015.
- [58] SCISSOR. SCISSOR project website, Available in <http://scissor-project.com/>, Accessed in December 2015.
- [59] K. Davis, C. Davis, S. Zonouz, R. Bobba, R. Berthier, L. Garcia, and P. Sauer. A cyber-physical modeling and assessment framework for power grid infrastructures. In *IEEE Transactions on Smart Grid*, September 2015.
- [60] R. Liu, C. Vellaithurai, S.. Biswas, T. Gamage, and A.. Srivastava. Analyzing the cyber-physical impact of cyber events on the power grid. In *IEEE Transaction on Smart Grids*, September 2015.
- [61] N. Medeiros. *A fault-and intrusion-tolerant architecture for EDP Distribuição SCADA system*. PhD thesis, 2011.
- [62] 'flexibility'. Merriam-webster online dictionary, Available in <http://www.merriam-webster.com/dictionary/flexibility>, Accessed in September 2010.
- [63] S. Goldman, R. Nagel, and K. Preiss. *Agile competitors and virtual organizations: strategies for enriching the costumer*. Van Nostrand Reinhold, New York, 1995.
- [64] P. Winston and K. Prendergast. *The AI business: The commercial uses of artificial intelligence*. 1984.
- [65] J. Barata and M. Onori. Evolvable assembly and exploiting emergent behaviour. In *IEEE International Symposium on Industrial Electronics*, 2006.
- [66] SPARKS. D2.2 threat and risk assessment methodology, September 2015.
- [67] CEN-CENELEC-ETSI Smart Grid Coordination Group. Smart grid reference architecture, November 2012.
- [68] SPARKS. D2.3 tools for smart grid cyber security, March 2016.
- [69] RDF Working Group. RDF standards, Available in http://www.w3.org/standards/techs/rdf#w3c_all, Accessed in October 2016.
- [70] OWL Working Group. OWL recommendation, Available in <http://www.w3.org/TR/owl-features/>, Accessed in October 2016.
- [71] D. Kosutic. Catalogue of threats & vulnerabilities, Available at <http://advisera.com/27001academy/knowledgebase/threats-vulnerabilities/>, Accessed in January 2016.

- [72] ISO/IEC. ISO/IEC 27002:2013, Information technology - Security techniques - Code of practice for information security controls, 2013.
- [73] ISO/IEC. ISO/IEC 27001:2013, Information technology - Security techniques - Information security management systems - Requirements, 2013.
- [74] M. Wooldridge. *An Introduction to MultiAgent Systems*. Second edition edition, 2009.
- [75] FIPA. FIPA ACL message structure specification, Available in <http://www.fipa.org/specs/fipa00061/SC00061G.html>, Accessed in October 2010.
- [76] FIPA. FIPA request interaction protocol specification, Available in <http://www.fipa.org/specs/fipa00026/SC00026H.html>, Accessed in October 2010.
- [77] FIPA. FIPA contract net interaction protocol specification, Available in <http://www.fipa.org/specs/fipa00029/SC00029H.html>, Accessed in October 2010.
- [78] FIPA. FIPA subscribe interaction protocol specification, Available in <http://www.fipa.org/specs/fipa00035/SC00035H.html>, Accessed in October 2010.
- [79] OWASP. OWASP secure coding practice quick reference guide, November 2010.
- [80] SWI-Prolog. SWI-Prolog - Robust, mature, free. Prolog for the real world., Available in <http://www.swi-prolog.org/>, Accessed in September 2016.
- [81] P. Blackburn, J. Bos, and K. Striegnitz. Learn prolog now!, Available in <http://www.learnprolognow.org/>, Accessed in September 2016.
- [82] M. Zillgith. Open source library for IEC 61850, Available in <http://libiec61850.com/libiec61850/>, Accessed in June 2016.
- [83] Fraunhofer ISE. Open MUC, Available in <https://www.openmuc.org/iec-61850/>, Accessed in June 2016.
- [84] syslog4j.org. Syslog4j: Complete syslog implementation for java, Available in <http://www.syslog4j.org/>, Accessed in June 2016.
- [85] snmp4j.org. The SNMP API for Java, Available in <http://www.snmp4j.org/>, Accessed in June 2016.
- [86] JCraft. Jsch - Java secure channel, Available in <http://www.jcraft.com/jsch/>, Accessed in June 2016.

Appendix A

Table of Threats, Vulnerabilities and Controls

Table A.1: Threat and vulnerability assessment.

ID	Threat	Vulnerability	Control
A01	Authentication failure	Inadequate last resort authentication	Local credentials fall-back
A02	Corrupted data or configurations	Inadequate error detection or correction	Error detection and correction mechanisms
A03	Facility destruction	Inadequate tolerance capability	Geographically redundant application instances
A04	Facility services interruption	Inadequate tolerance capability	Geographically redundant application instances
A05	Faulty OS	Inadequate tolerance capability	Redundant application instances in different VMs or physical hosts
A06	Faulty communication channel	Inadequate data validation measures	Error detection and correction mechanisms
A07	Faulty configurations	Inadequate massive deployment capability	Configuration deployment systems
A08	Faulty configurations	Inadequate user input validation	Automatic input and workflow restriction, validation and confirmation
A09	Faulty control device	Inadequate command confirmation	Command success confirmation
A10	Faulty data source	Inadequate data validation measures	Data comparison between redundant data sources

Continued on next page

Table A.1 – *Continued from previous page*

ID	Threat	Vulnerability	Control
A11	Faulty data source or control device	Inadequate redundancy measures	Redundant data sources and control devices
A12	Faulty hardware	Inadequate tolerance capability	Redundant application instances in different physical hosts
A13	Faulty hypervisor	Inadequate tolerance capability	Redundant application instances in different physical hosts
A14	Faulty user or permission assignment	Inadequate massive deployment capability	Centralized user accounts
A15	Faulty user or permission assignment	Inadequate permission levels or granularity	Fine-grained RBAC
A16	Loss of data or configurations	Inadequate backups	Regular backups
A17	Malicious code	Inadequate recovery capability	Proactive and reactive system recovery
A18	Man-in-the-middle	Inadequate authentication measures	Secure authentication protocols
A19	Network partitions	Inadequate application distribution	Distributed application architecture
A20	Outdated smart grid data	Inadequate smart grid data update	Regular proactive data collection and update
A21	Shared user accounts	Inadequate accountability capability	Personal accounts
A22	Software errors	Inadequate massive deployment capability	Update deployment systems
A23	Software errors	Inadequate requirement specification	Secure coding practices specification requirement
A24	Unauthorized application access	Inadequate authentication and authorization	User accounts
A25	Unauthorized application access	Inadequate security event monitoring	Security logging and monitoring
A26	Unauthorized communication change	Inadequate data validation measures	Secure hashing algorithms and hash comparison
A27	Unauthorized data or configuration change	Inadequate access control	RBAC
A28	Unauthorized data or configuration change	Inadequate security event monitoring	Security logging and monitoring

Continued on next page

Table A.1 – *Continued from previous page*

ID	Threat	Vulnerability	Control
A29	Unauthorized data or configuration change	Inadequate user input validation	Automatic input and workflow restriction, validation and confirmation
A30	Unintentional data or configuration change	Inadequate security event monitoring	Security logging and monitoring
A31	Unintentional data or configuration change	Inadequate user input validation	Automatic input and workflow restriction, validation and confirmation
A32	User account sharing	Inadequate idle session management	Automatic idle user logout
A33	Weak user account passwords	Inadequate password management	Strong password policy
B01	Faulty OS	Inadequate health and performance monitoring	Health and performance monitoring
B02	Faulty communication channel	Inadequate health and performance monitoring	Health and performance monitoring systems
B03	Faulty communication channel	Inadequate redundancy measures	Redundant communications channels
B04	Faulty configurations	Inadequate change management	Regular change management policy review
B05	Faulty configurations	Inadequate configuration procedures	Regular configuration procedures review
B06	Faulty configurations	Inadequate support	Manufacturer and/or vendor contractual support
B07	Faulty configurations	Inadequate testing	Regular penetration testing
B08	Faulty configurations	Lack of knowledge and awareness	User training and awareness
B09	Faulty data source or control device	Inadequate health and performance monitoring	Health and performance monitoring
B10	Faulty facility services	Inadequate tolerance capability	Redundant services supply
B11	Faulty hardware	Inadequate health and performance monitoring	Health and performance monitoring
B12	Faulty hypervisor	Inadequate health and performance monitoring	Health and performance monitoring
B13	Faulty user or permission assignment	Inadequate user management policy	Regular user management policy review
B14	Faulty user or permission assignment	Inadequate user or permission review	Regular user account and permission review

Continued on next page

Table A.1 – Continued from previous page

ID	Threat	Vulnerability	Control
B15	Lack of HR resources	Inadequate HR management	Regular HR capacity review
B16	Lack of logical resources	Inadequate capacity management	Regular capacity management policy review
B17	Lack of physical resources	Inadequate capacity management	Regular capacity management policy review
B18	Loss of data or configurations	Inadequate storage redundancy	Data store redundancy
B19	Malicious code	Inadequate anti-malware protection	Whitelisting, antivirus, host-based firewall, intrusion detection and prevention
B20	Physical malicious intrusion	Inadequate physical security monitoring	Physical security logging and monitoring
B21	Software errors	Inadequate testing	Penetration testing and vulnerability management
B22	Software errors	Inadequate update procedures	Regular update procedures review
B23	Unauthorized application access	Inadequate intrusion detection and prevention	Host and network-based firewall, intrusion detection and prevention
B24	Unauthorized application access	Inadequate network segmentation	Physical , VPN, VLAN and VRF configurations
B25	Unintentional data or configuration change	Lack of knowledge and awareness	User training and awareness
B26	User account sharing	Lack of security awareness	Security awareness training
B27	Weak security implementation	Inadequate robustness testing	Robustness testing and evaluation
C01	Communication eavesdropping	Inadequate communications encryption	Application level communications encryption
C02	Communication eavesdropping	Inadequate key exchange	Secure key exchange protocols
C03	Faulty application behavior	Inadequate recovery capability	Proactive and reactive system recovery
C04	Faulty application behavior	Inadequate tolerance capability	Replicated application instances with active replication protocol
C05	Hardware theft	Inadequate encryption keys storage	Cryptographic key vault
C06	Hardware theft	Inadequate storage encryption	Application or OS level storage encryption

Continued on next page

Table A.1 – *Continued from previous page*

ID	Threat	Vulnerability	Control
C07	Malicious application behavior	Inadequate recovery capability	Proactive and reactive system recovery
C08	Malicious application behavior	Inadequate security event monitoring	Security logging and monitoring
C09	Malicious application behavior	Inadequate tolerance capability	Replicated application instances with active replication protocol
C10	Message replay	Inadequate message validation measures	Secure message creation and flow validation
C11	Unauthorized data access	Inadequate encryption keys storage	Cryptographic key vault
C12	Unauthorized data access	Inadequate storage encryption	Application or OS level storage encryption
D01	Application unavailability	Inadequate business continuity capability	Regular business continuity plan review
D02	Data leakage	Inadequate data monitoring	DLP systems
D03	Data leakage	Inadequate equipment disposal	Equipment disposal procedures
D04	Data leakage	Inadequate information classification	Regular information classification policy review
D05	Data leakage	Lack of knowledge and awareness	User training and awareness
D06	Information theft	Inadequate HR recruitment and selection	HR recruitment and selection criteria
D07	Malicious application behavior	Inadequate anti-malware protection	Whitelisting, antivirus, host-based firewall, intrusion detection and prevention
D08	Malicious application behavior	Inadequate incident response capability	Regular incident response policy review

Appendix B

POC Knowledge

This chapter includes the complete content of the knowledge base that was implemented for the Proof of Concept (POC).

B.1 Facts

Figures B.1 to B.10 depict the knowledge facts.

B.2 Rules

Figures B.11 to B.26 depict the knowledge rules.

```
% component(ComponentID, ComponentType)
component(rt1, com:router).
  % interface(ComponentID, InterfaceID, InterfaceType)
  interface(rt1, rt1:core, core).
  interface(rt1, rt1:eth1, ethernet).
  interface(rt1, rt1:eth2, ethernet).

  % connected(Service, Interface1ID, Interface2ID,
  %   Dependencies, Interface1Controls, Interface2Controls)
  connected(communications, scada1:app1, rt1:eth2, [], [], []).
  connected(communications, rt1:eth2, rt1:core,
    [], [firewall, routing], []).
  connected(communications, rt1:core, rt1:eth1,
    [], [], [routing, firewall]).
  connected(communications, rt1:eth1, sc1:eth2, [], [], []).

  connected(scada, scada1:app1, sc1:app1, [communications], [], []).

  % opposite(ControlType, State, OppositeState)
  opposite-fwRule, allow, deny).
  opposite-fwRule, deny, allow).

  % currentState(InterfaceID, Control, ControlParameters)
  currentState(rt1:eth1, route, (sc1:app1)).
  currentState(rt1:eth2, route, (scada1:app1)).

  currentState(rt1:eth1, fwRule, (allow, scada1:app1, sc1:app1)).
  currentState(rt1:eth2, fwRule, (allow, scada1:app1, sc1:app1)).
```

Figure B.1: Router 1 facts.

```
component(sc1, scada:sc).
  interface(sc1, sc1:eth1, ethernet).
  interface(sc1, sc1:eth2, ethernet).
  interface(sc1, sc1:app1, application).

  connected(communications, sc1:eth2, sc1:app1,
    [], [scmode, firewall, routing], []).
  connected(communications, sc1:eth1, sc1:app1,
    [], [scmode, firewall, routing], []).
  connected(communications, sc1:app1, sc1:eth1,
    [], [scmode], [routing, firewall]).
  connected(communications, sc1:eth1, sw2:eth2,
    [], [], []).

  connected(scada, sc1:app1, ied9:app1, [communications], [], []).

  opposite(scmode, local, remote).
  opposite(scmode, remote, local).

  currentState(sc1:eth1, route, (hmil:app1)).
  currentState(sc1:eth1, route, (ied9:app1)).
  currentState(sc1:eth2, route, (scada1:app1)).

  currentState(sc1:eth1, fwRule, (allow, sc1:app1, ied9:app1)).
  currentState(sc1:eth1, fwRule, (allow, hmil:app1, sc1:app1)).
  currentState(sc1:eth2, fwRule, (allow, scada1:app1, sc1:app1)).

  policy(sc1:app1, scmode, (local)).

  currentState(sc1:app1, scmode, (local)).
```

Figure B.2: Substation controller 1 facts.

```

component(hmil, scada:hmi).
  interface(hmil, hmil:eth1, ethernet).
  interface(hmil, hmil:os1, os).
  interface(hmil, hmil:appl, application).
  interface(hmil, hmil:phyl, physical).

  connected(communications, hmil:phyl, hmil:os1,
    [], [], [authentication]).
  connected(communications, hmil:os1, hmil:appl,
    [], [], [authentication]).
  connected(communications, hmil:appl, hmil:eth1,
    [], [], [routing, firewall]).
  connected(communications, hmil:eth1, sw2:eth3, [], [], []).

  connected(scada, hmil:phyl, hmil:appl, [communications], [], []).
  connected(scada, hmil:appl, sc1:appl, [communications], [], []).

  % policy(InterfaceID, Control, Parameters)
  policy(hmil:os1, user, (alice)).
  policy(hmil:appl, user, (alice)).

  currentState(hmil:eth1, route, (sc1:appl)).

  currentState(hmil:eth1, fwRule, (allow, hmil:appl, sc1:appl)).

  currentState(hmil:os1, user, (alice)).
  currentState(hmil:appl, user, (alice)).

```

Figure B.3: HMI 1 facts.

```

component(sw2, com:switch).
  interface(sw2, sw2:core, core).
  interface(sw2, sw2:eth1, ethernet).
  interface(sw2, sw2:eth2, ethernet).
  interface(sw2, sw2:eth3, ethernet).

  connected(communications, sw2:eth3, sw2:core,
    [], [switching], []).
  connected(communications, sw2:core, sw2:eth2,
    [], [], [switching]).
  connected(communications, sw2:eth2, sc1:eth1, [], [], []).
  connected(communications, sw2:eth2, sw2:core,
    [], [switching], []).
  connected(communications, sw2:core, sw2:eth1,
    [], [], [switching]).
  connected(communications, sw2:eth1, fw1:eth2, [], [], []).

  currentState(sw2:eth1, vlan, (sw2:vlan1)).
  currentState(sw2:eth2, vlan, (sw2:vlan1)).
  currentState(sw2:eth3, vlan, (sw2:vlan1)).

```

Figure B.4: Switch 2 facts.

```
component(fw1, com:firewall).
  interface(fw1, fw1:core, core).
  interface(fw1, fw1:eth1, ethernet).
  interface(fw1, fw1:eth2, ethernet).

  connected(communications, fw1:eth2, fw1:core,
    [], [firewall, routing], []).
  connected(communications, fw1:core, fw1:eth1,
    [], [], [routing, firewall]).
  connected(communications, fw1:eth1, sw1:eth2, [], [], []).

  currentState(fw1:eth1, route, (ied9:appl)).
  currentState(fw1:eth2, route, (sc1:appl)).

  currentState(fw1:eth1, fwRule, (allow, sc1:appl, ied9:appl)).
  currentState(fw1:eth2, fwRule, (allow, sc1:appl, ied9:appl)).
```

Figure B.5: Firewall 1 facts.

```
component(sw1, com:switch).
  interface(sw1, sw1:core, core).
  interface(sw1, sw1:eth1, ethernet).
  interface(sw1, sw1:eth2, ethernet).

  connected(communications, sw1:eth2, sw1:core,
    [], [switching], []).
  connected(communications, sw1:core, sw1:eth1,
    [], [], [switching]).
  connected(communications, sw1:eth1, ied9:eth1, [], [], []).

  currentState(sw1:eth1, vlan, (sw1:vlan1)).
  currentState(sw1:eth2, vlan, (sw1:vlan1)).
```

Figure B.6: Switch 1 facts.

```

component(ied9, scada:ied).
  interface(ied9, ied9:ser1, serial).
  interface(ied9, ied9:eth1, ethernet).
  interface(ied9, ied9:appl, application).
  interface(ied9, ied9:phy1, physical).

  connected(communications, ied9:eth1, ied9:appl,
    [], [iedmode], []).
  connected(communications, ied9:appl, ied9:ser1,
    [], [iedmode], []).
  connected(communications, ied9:ser1, es9:ser1,
    [], [], []).
  connected(communications, ied9:phy1, ied9:appl,
    [], [iedmode], []).

  connected(scada, ied9:phy1, ied9:appl,
    [communications], [], []).
  connected(scada, ied9:appl, es9:me1,
    [communications], [], []).

  opposite(iedmode, local, remote).
  opposite(iedmode, remote, local).

  policy(ied9:appl, iedmode, (remote)).

  currentState(ied9:appl, iedmode, (remote)).

```

Figure B.7: IED 9 facts.

```

component(es9, scada:es).
  interface(es9, es9:el1, electrical).
  interface(es9, es9:el2, electrical).
  interface(es9, es9:me1, mechanism).
  interface(es9, es9:ser1, serial).
  interface(es9, es9:phy1, physical).

  connected(communications, es9:ser1, es9:me1, [], [], []).

  connected(mechanical, es9:phy1, es9:me1, [], [], []).

  connected(electrical, es9:me1, es9:el2, [], [position], []).
  connected(electrical, es9:el1, es9:me1, [], [], []).

  connected(scada, es9:phy1, es9:me1, [mechanical], [], []).
  connected(scada, es9:me1, es9:el2, [electrical], [], []).

  opposite(position, closed, opened).
  opposite(position, opened, closed).

  % command(Control, CommandID, ExpectedConsequence)
  command(position, close, closed).
  command(position, open, opened).

  policy(es9:me1, position, (closed)).

  currentState(es9:me1, position, (closed)).

```

Figure B.8: Electrical switch 9 facts.

```
component(dol, physical:door).
  interface(dol, dol:phy1, physical).
  interface(dol, dol:phy2, physical).

  connected(physical, dol:phy2, dol:phy1,
    [], [authentication], []).
  connected(physical, dol:phy1, hmil:phy1, [], [], []).
  connected(physical, dol:phy1, ied9:phy1, [], [], []).
  connected(physical, dol:phy1, es9:phy1, [], [], []).

  connected(scada, dol:phy2, hmil:phy1, [physical], [], []).
  connected(scada, dol:phy2, ied9:phy1, [physical], [], []).
  connected(scada, dol:phy2, es9:phy1, [physical], [], []).

  policy(dol:phy2, user, (alice)).

  currentState(dol:phy2, user, (alice)).

component(scada1, shee2:component).
  interface(scada1, scada1:appl, application).
```

Figure B.9: Door 1 facts.

```
component(scada1, shee2:component).
  interface(scada1, scada1:appl, application).

% shee (SHEEID)
shee(shee2).
```

Figure B.10: SHEE2 facts as seen from SHEE1.

```

% checkControl(Control, FlowDirection, InterfaceID,
%   User, FlowInformation, Configuration)
checkControl(switching, _, A, User, FlowInfo, Config) :-
    currentState(A, vlan, Parameters),
    Config = [(A, vlan, Parameters)].

% monitorConfig(Interface1Configuration, Interface2Configuration)
monitorConfig((A1, vlan, (Vlan1)), (A2, vlan, (Vlan2))) :-
    interface(Component, A1, _),
    interface(Component, A2, _),
    Vlan2 == Vlan1.

% diagnoseConfig(Interface1Configuration, Interface2Configuration,
%   DiagnosisResult)
diagnoseConfig((A1, vlan, (Vlan1)),
    (A2, vlan, (Vlan2)), Diagnostic) :-
    interface(Component, A1, _),
    interface(Component, A2, _),

    ((Vlan2 == Vlan1,
    Diagnostic = [(A2, okVlan, (Vlan2))]);

    (\+(Vlan2 == Vlan1),
    Diagnostic = [(A2, wrongVlan, (Vlan2, Vlan1))])).

% recoverConfig(DiagnosisResult, RecoveryPlan)
recoverConfig((A, okVlan, Parameters), []).

recoverConfig((A, wrongVlan, (Vlan2, Vlan1)),
    [(A, changeVlanTo, (Vlan1))]).

```

Figure B.11: Switching rules.

```

checkControl(routing, FlowDir, A,
    User, (Source, Target), Config) :-
    (FlowDir == in,
    Config = [(A, route, (Source))]);

    (FlowDir == out,
    Config = [(A, route, (Target))]).

% monitorConfig(Interface1Configuration)
monitorConfig((A, route, Parameters)) :-
    currentState(A, route, Parameters).

% diagnoseConfig(Interface1Configuration, DiagnosisResult)
diagnoseConfig((A, route, Parameters), Diagnostic) :-
    (currentState(A, route, Parameters),
    Diagnostic = [(A, okRoute, Parameters)]);

    (\+currentState(A, route, Parameters),
    Diagnostic = [(A, noRoute, Parameters)]).

recoverConfig((A, okRoute, Parameters), []).

recoverConfig((A, noRoute, Parameters),
    [(A, addRoute, Parameters)]).

```

Figure B.12: Routing rules.


```
checkControl(firewall, _, A, User, (Source, Target), Config) :-
    Config = [(A, fwRule, (allow, Source, Target))].

monitorConfig((A, fwRule, Parameters)) :-
    currentState(A, fwRule, Parameters).

diagnoseConfig((A, fwRule, (Permission, Source, Target)),
    Diagnostic) :-
    (currentState(A, fwRule, (Permission, Source, Target)),
    Diagnostic = [(A, okFwRule, (Permission, Source, Target))]);

    (opposite(fwRule, Permission, OP),
    currentState(A, fwRule, (OP, Source, Target)),
    Diagnostic = [(A, wrongPermission,
        (Permission, Source, Target))]);

    (\+currentState(A, fwRule, (Permission, Source, Target)),
    opposite(fwRule, Permission, OP),
    \+currentState(A, fwRule, (OP, Source, Target)),
    Diagnostic = [(A, noFwRule, (Permission, Source, Target))]).

recoverConfig((A, okFwRule, Parameters), []).

recoverConfig((A, wrongPermission, Parameters),
    [(A, changeFwRule, Parameters)]).

recoverConfig((A, noFwRule, Parameters),
    [(A, addFwRule, Parameters)]).
```

Figure B.13: Firewall rules.

```

checkControl(authentication, _, A, User, FlowInfo, Config) :-
    Config = [(A, user, (User))].

monitorConfig((A, user, Parameters)) :-
    policy(A, user, Parameters),
    currentState(A, user, Parameters).

diagnoseConfig((A, user, Parameters), Diagnostic) :-
    (policy(A, user, Parameters),
     currentState(A, user, Parameters),
     Diagnostic = [(A, okUser, Parameters)]);

    (\+policy(A, user, Parameters),
     currentState(A, user, Parameters),
     Diagnostic = [(A, userNAByPol, Parameters)]);

    (policy(A, user, Parameters),
     \+currentState(A, user, Parameters),
     Diagnostic = [(A, userNotConfigured, Parameters)]);

    (\+policy(A, user, Parameters),
     \+currentState(A, user, Parameters),
     Diagnostic = [(A, userNAPolNConf, Parameters)]).

recoverConfig((A, okUser, Parameters), []).

recoverConfig((A, userNAByPol, Parameters),
              [(A, removeUser, Parameters)]).

recoverConfig((A, userNotConfigured, Parameters),
              [(A, addUser, Parameters)]).

recoverConfig((A, userNAPolNConf, Parameters),
              [(A, usersLimByPolicy)]).

```

Figure B.14: Authentication rules.

```

checkControl(scmode, _, A, User, (Source, Target), Config) :-
    (interface(_, A, ethernet),
     interface(Component, Source, _),
     component(Component, scada:hmi),
     Config = [(A, scmode, (local))]);

    (interface(_, A, ethernet),
     interface(Component, Source, _),
     \+component(Component, scada:hmi),
     Config = [(A, scmode, (remote))]);

    (interface(_, A, application),
     currentState(A, scmode, Parameters),
     Config = [(A, scmode, Parameters)]).

monitorConfig((A1, scmode, Parameters1),
              (A2, scmode, Parameters2)) :-
    policy(A2, scmode, Parameters1),
    currentState(A2, scmode, Parameters1).

diagnoseConfig((A1, scmode, (Model)),
               (A2, scmode, (Mode2)), Diagnostic) :-
    (policy(A2, scmode, (Model)),
     currentState(A2, scmode, (Model)),
     Diagnostic = [(A2, okScmode, (Model))]);

    (opposite(scmode, Model, OM),
     policy(A2, scmode, (OM)),
     currentState(A2, scmode, (Model)),
     Diagnostic = [(A2, scmodeNAByPol, (Model, OM))]);

    (policy(A2, scmode, (Model)),
     opposite(scmode, Model, OM),
     currentState(A2, scmode, (OM)),
     Diagnostic = [(A2, scmodeNotConfigured, (Model, OM))]);

    (opposite(scmode, Model, OM),
     policy(A2, scmode, (OM)),
     currentState(A2, scmode, (OM)),
     Diagnostic = [(A2, scmodeNAPolNConf, (Model))]).

recoverConfig((A, okScmode, Parameters), []).

recoverConfig((A, scmodeNAByPol, (Mode, OM)),
              [(A, changeScmodeTo, OM)]).

recoverConfig((A, scmodeNotConfigured, (Mode, OM)),
              [(A, changeScmodeTo, Mode)]).

recoverConfig((A, scmodeNAPolNConf, Parameters),
              [(A, scmodeLimByPolicy)]).

```

Figure B.15: Substation controller operating mode rules.

```

checkControl(iedmode, _, A, User, (Source, Target), Config) :-
    (interface(Component, A, physical),
     Config = [(A, iedmode, (local))]);

    (interface(_, A, ethernet),
     Config = [(A, iedmode, (remote))]);

    (interface(_, A, application),
     currentState(A, iedmode, Parameters),
     Config = [(A, iedmode, Parameters)]).

monitorConfig((A1, iedmode, Parameters1),
              (A2, iedmode, Parameters2)) :-
    policy(A2, iedmode, Parameters1),
    currentState(A2, iedmode, Parameters1).

diagnoseConfig((A1, iedmode, (Model)), (A2, iedmode, (Mode2)),
              Diagnostic) :-
    (policy(A2, iedmode, (Model)),
     currentState(A2, iedmode, (Model)),
     Diagnostic = [(A2, okIedmode, (Model))]);

    (opposite(iedmode, Model, OM),
     policy(A2, iedmode, (OM)),
     currentState(A2, iedmode, (Model)),
     Diagnostic = [(A2, iedmodeNAByPol, (Model, OM))]);

    (policy(A2, iedmode, (Model)),
     opposite(iedmode, Model, OM),
     currentState(A2, iedmode, (OM)),
     Diagnostic = [(A2, iedmodeNotConfigured, (Model, OM))]);

    (opposite(iedmode, Model, OM),
     policy(A2, iedmode, (OM)),
     currentState(A2, iedmode, (OM)),
     Diagnostic = [(A2, iedmodeNAPolNConf, (Model))]).

recoverConfig((A, okIedmode, Parameters), []).

recoverConfig((A, iedmodeNAByPol, (Mode, OM)),
              [(A, changeIedmodeTo, OM)]).

recoverConfig((A, iedmodeNotConfigured, (Mode, OM)),
              [(A, changeIedmodeTo, Mode)]).

recoverConfig((A, iedmodeNAPolNConf, Parameters),
              [(A, iedmodeLimByPolicy)]).

```

Figure B.16: IED operating mode rules.

```

checkControl(position, _, A, User, FlowInfo, Config) :-
    currentState(A, position, State),
    Config = [(A, position, State)].

monitorConfig((A, position, Parameters)) :-
    policy(A, position, Parameters),
    currentState(A, position, Parameters).

diagnoseConfig((A, position, (State)), Diagnostic) :-
    (policy(A, position, (State)),
     currentState(A, position, (State))),
    Diagnostic = [(A, positionOk, (State))]);

    (opposite(position, State, OS),
     policy(A, position, (OS)),
     currentState(A, position, (State))),
    Diagnostic = [(A, positionNAByPol, (State, OS))]);

    (policy(A, position, (State)),
     opposite(position, State, OS),
     currentState(A, position, (OS))),
    Diagnostic = [(A, positionNotConfigured, (State, OS))]);

    (opposite(position, State, OS),
     policy(A, position, (OS)),
     currentState(A, position, (OS))),
    Diagnostic = [(A, positionNAPolNConf, (State))]).

recoverConfig((A, positionOk, Parameters), []).

recoverConfig((A, positionNAByPol, (State, OS)),
              [(A, Command, OS)]) :-
    command(position, Command, OS).

recoverConfig((A, positionNotConfigured, (State, OS)),
              [(A, changePositionTo, State)]) :-
    command(position, Command, State).

recoverConfig((A, positionNAPolNConf, Parameters),
              [(A, positionLimByPolicy)]).

```

Figure B.17: Electrical switch rules.

```

% checkControls(InterfaceID, FlowDirection, ControlList,
% User, FlowInformation, ConfigurationList)
checkControls(A, FlowDir, ControlList,
              User, FlowInfo, ConfigList) :-
    (ControlList == [],
     ConfigList = []);

    ([Control|Rest] = ControlList,
     checkControl(Control, FlowDir, A, User, FlowInfo, Config),
     checkControls(A, FlowDir, Rest, User, FlowInfo, ConfigList1),
     myAppend([Config, ConfigList1], ConfigList)).

```

Figure B.18: Rule to retrieve the configurations associated with a component interface.

```

% checkSupport(InterfaceA, InterfaceB, ServiceList,
%   User, Path, ConfigurationList)
checkSupport(A, B, List, User, Path, ConfigList) :-
    (List == [],
    Path = [],
    ConfigList = []);

    ([Service|Rest] = List,
    checkService(Service, A, B,
        User, (A, B), [A], Path1, ConfigList1),
    checkSupport(A, B, Rest, User, Path2, ConfigList2),
    myAppend([Path2, Path1, [A]], Path),
    myAppend([ConfigList2, ConfigList1], ConfigList)).

```

Figure B.19: Rule to cycle through the service dependencies associated with a connection.

```

% checkService(ServiceID, Interface1ID, Interface2ID,
%   User, FlowInformation, VisitedInterfaces,
%   Path, ConfigurationList)
checkService(Service, A, B,
    User, FlowInfo, Visited, Path, ConfigList) :-
    (connected(Service, A, B,
        DepList, ControlListA, ControlListB),
    checkControls(A, in, ControlListA,
        User, FlowInfo, ConfigListA),
    checkSupport(A, B, DepList,
        User, Path1, ConfigList1),
    checkControls(B, out, ControlListB,
        User, FlowInfo, ConfigListB),
    myAppend([B], Path1, Path),
    myAppend([ConfigListB, ConfigList1, ConfigListA],
        ConfigList));

    (connected(Service, A, C,
        DepList, ControlListA, ControlListB),
    checkControls(A, in, ControlListA,
        User, FlowInfo, ConfigListA),
    checkSupport(A, C,
        DepList, User, Path1, ConfigList1),
    checkControls(C, out, ControlListB,
        User, FlowInfo, ConfigListB),
    C \== B,
    \+member(C, Visited),
    checkService(Service, C, B,
        User, FlowInfo, [C|Visited], Path2, ConfigList2),
    myAppend([Path2, [C], Path1], Path),
    myAppend([ConfigList2, ConfigListB, ConfigList1,
        ConfigListA], ConfigList)).

```

Figure B.20: Rule to cycle through the connections associated with a service.

```

% check(ServiceID, InterfacelID, Interface2ID,
% User, Path, ConfigurationList)
check(Service, A, B, User, Path, ConfigList) :-
    checkService(Service, A, B,
        User, (A, B), [A], Path1, ConfigList1),
    reverse(Path1, Path2),
    myAppend([[A], Path2], Path),
    reverse(ConfigList1, ConfigList).

```

Figure B.21: Rule to retrieve the configurations associated with a service.

```

% monitorConfigs(ConfigurationList)
monitorConfigs(ConfigList) :-
    (ConfigList == []);

    (ConfigList = [(A1, vlan, Parameter1),
        (A2, vlan, Parameter2)|Rest],
    monitorConfig((A1, vlan, Parameter1),
        (A2, vlan, Parameter2)),
    monitorConfigs(Rest));

    (ConfigList = [(A1, scmode, Parameter1),
        (A2, scmode, Parameter2)|Rest],
    monitorConfig((A1, scmode, Parameter1),
        (A2, scmode, Parameter2)),
    monitorConfigs(Rest));

    (ConfigList = [(A1, iedmode, Parameter1),
        (A2, iedmode, Parameter2)|Rest],
    monitorConfig((A1, iedmode, Parameter1),
        (A2, iedmode, Parameter2)),
    monitorConfigs(Rest));

    (ConfigList = [(A, Control, Parameters)|Rest],
    \+(Control == vlan),
    \+(Control == scmode),
    \+(Control == iedmode),
    monitorConfig((A, Control, Parameters)),
    monitorConfigs(Rest)).

% monitor(ServiceID, InterfacelID, Interface2ID, User)
monitor(Service, A, B, User) :-
    checkService(Service, A, B,
        User, (A, B), [A], _, ConfigList1),
    reverse(ConfigList1, ConfigList2),
    monitorConfigs(ConfigList2).

```

Figure B.22: Rules associated with the monitoring activity.

```

% diagnoseConfigs(ConfigurationList, DiagnosisResult)
diagnoseConfigs(ConfigList, Diagnostic) :-
    (ConfigList == [],
     Diagnostic = []);

    (ConfigList = [(A1, vlan, Parameter1),
                  (A2, vlan, Parameter2)|Rest],
     diagnoseConfig((A1, vlan, Parameter1),
                   (A2, vlan, Parameter2), DiagnosticA),
     diagnoseConfigs(Rest, DiagnosticR),
     myAppend([DiagnosticA, DiagnosticR], Diagnostic));

    (ConfigList = [(A1, scmode, Parameter1),
                  (A2, scmode, Parameter2)|Rest],
     diagnoseConfig((A1, scmode, Parameter1),
                   (A2, scmode, Parameter2), DiagnosticA),
     diagnoseConfigs(Rest, DiagnosticR),
     myAppend([DiagnosticA, DiagnosticR], Diagnostic));

    (ConfigList = [(A1, iedmode, Parameter1),
                  (A2, iedmode, Parameter2)|Rest],
     diagnoseConfig((A1, iedmode, Parameter1),
                   (A2, iedmode, Parameter2), DiagnosticA),
     diagnoseConfigs(Rest, DiagnosticR),
     myAppend([DiagnosticA, DiagnosticR], Diagnostic));

    (ConfigList = [(A, Control, Parameter1)|Rest],
     \+(Control == vlan),
     \+(Control == scmode),
     \+(Control == iedmode),
     diagnoseConfig((A, Control, Parameter1), DiagnosticA),
     diagnoseConfigs(Rest, DiagnosticR),
     myAppend([DiagnosticA, DiagnosticR], Diagnostic)).

% diagnose(ServiceID, Interface1ID, Interface2ID,
% User, DiagnosisResult)
diagnose(Service, A, B, User, Diagnostic) :-
    checkService(Service, A, B,
                 User, (A, B), [A], _, ConfigList1),
    reverse(ConfigList1, ConfigList2),
    diagnoseConfigs(ConfigList2, Diagnostic).

```

Figure B.23: Rules associated with the diagnosis activity.


```

% recoverConfigs(DiagnosisList, RecoveryPlan)
recoverConfigs(DiagList, RecoveryPlan) :-
    (DiagList == [],
     RecoveryPlan = []);

    (DiagList = [Diagnostic|Rest],
     recoverConfig(Diagnostic, ConfigA),
     recoverConfigs(Rest, PlanR),
     myAppend([ConfigA, PlanR], RecoveryPlan)).

% recover(ServiceID, Interface1ID, Interface2ID,
% User, RecoveryPlan)
recover(Service, A, B, User, RecoveryPlan) :-
    checkService(Service, A, B,
                 User, (A, B), [A], _, ConfigList1),
    reverse(ConfigList1, ConfigList2),
    diagnoseConfigs(ConfigList2, Diagnostic),
    recoverConfigs(Diagnostic, RecoveryPlan).

```

Figure B.24: Rules associated with the recovery activity.

```

% myAppend(ListOfLists, ConcatenationResult)
myAppend(ListOfLists, List) :-
    (ListOfLists == [],
     List = []);

    (ListOfLists = [First|Rest],
     myAppend(Rest, List1),
     append(First, List1, List)).

```

Figure B.25: Rule to concatenate a list of lists.

```

:- disjointous
   component/2,
   interface/3,
   connected/6,
   opposite/3,
   currentState/3,
   policy/3,
   checkControl/6,
   monitorConfig/1,
   monitorConfig/2,
   diagnoseConfig/2,
   diagnoseConfig/3,
   recoverConfig/2.

:- style_check(-singleton).

```

Figure B.26: Prolog initializations.