The MEDINA Controlled Natural Language

(Whitepaper)

Editor(s):	Marinella Petrocchi, Michela Fazzolari
Responsible Partner:	Consiglio Nazionale delle Ricerche (CNR)
Status-Version:	Final – v1.0
Date:	30.09.2023
Distribution level (CO, PU):	PU

(cc) BY-SA

Project Number:	952633
Project Title:	MEDINA

Editor(s):	Marinella Petrocchi, Michela Fazzolari (CNR)
Contributor(s):	
Reviewer(s):	Cristina Martínez (TECNALIA)
Approved by:	All Partners
Recommended readers:	Cloud Service Providers, Auditors, IT Security-related

Keyword List:	MEDINA Cloud Certification Language, MEDINA CNL, CNL
Licensing information:	This work is licensed under Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) http://creativecommons.org/licenses/by-sa/3.0/
Disclaimer	This document reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein.

Document Description

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	07.09.2023	First draft version	Marinella Petrocchi
v0.2	11.09.2023	Add MEDINA Template	Michela Fazzolari
v0.3	12.09.2023	Add Executive Summary	Michela Fazzolari
v0.4	14.09.2023	Add Benefits, Drawbacks and Conclusions	Michela Fazzolari
v1.0	30.09.2023	Final revised version	Cristina Martinez

(cc) BY-SA

Table of contents

List of figures

CC BY-SA

Terms and abbreviations

BNF	Backus-Naur Form
CNL	Controlled Natural Language
CNL4DSA	Controlled Natural Language for Data Sharing Agreement
CSP	Cloud Service Provider
EUCS	European Cybersecurity Certification Scheme for Cloud Services
GA	Grant Agreement
NL	Natural Language
XACML	eXtensible Access Control Markup Language

(cc) BY-SA

Executive Summary

This whitepaper provides an overview of the MEDINA Controlled Natural Language, which has been designed in the framework of the EU MEDINA project. This document highlights its pivotal role as a dedicated language designed to express requirements from schemes like the European Union Cloud Security Certification Scheme (EUCS) in a formal, machine-readable manner, to automate automatic compliance assessment for cybersecurity certification schemes.

The report initially introduces the motivations behind creating the Medina CNL to bridge the gap between natural language controls and automated compliance assessment. Then, it presents the structure and functionality of the Medina CNL, emphasizing its significance within the cloud service certification context. Furthermore, the benefits of using Medina CNL include enhanced automation, improved accuracy in compliance checks, and streamlined certification processes. The drawbacks of the presented approach are primarily related to the learning curve for users transitioning to CNL-based compliance assessments and potential limitations in expressing highly complex requirements.

In summary, the MEDINA CNL serves as a critical tool for automating the assessment of compliance with security certification schemes in cloud services. It bridges the gap between human-readable security requirements and machine-processing, enhancing the trust and security of cloud environments.

(cc) BY-SA

1 Introduction

In an increasingly digitized world, cybersecurity and data protection have become topmost priority. Regulatory bodies have established stringent requirements to safeguard sensitive information, particularly in cloud service environments. The adoption of cloud services involves a shift from direct management and oversight of data and applications to a more indirect form of control, raising concerns about matters such as security, privacy, transparency, and reliability. To enhance trust in cloud services, the adoption of Cloud Service certifications has emerged as an effective remedy.

Cloud Service certifications assure that Cloud Providers adhere to industry standards and best practices, by offering a certification to validate compliance with one or more security certification schemes. A security certification scheme, often referred to simply as a certification scheme, is a structured and standardized framework that defines a set of security requirements, controls, and guidelines that organizations or systems must adhere to in order to demonstrate their commitment to security best practices and compliance with established security standards.

However, requirements, controls and guidelines are often conveyed in Natural Language (NL), making them comprehensible to humans but challenging for automated systems to process and enforce. This challenge led the MEDINA consortium to define a dedicated language called the MEDINA Controlled Natural Language (CNL), which allows requirements defined in cybersecurity certification schemes, such as the European Union Cloud Security Certification Scheme (EUCS) [1], to be expressed in a formal, processable language.

EUCS controls, like other types of requirements, are written in natural language. While this allows Cloud Service Providers to understand the specific cloud security certification requirements, it poses a challenge because natural language is not machine-readable, making it impossible to automatically verify if controls and requirements are met. To assess compliance with these requirements in an automated and continuous manner, machine-readable input is needed, such as the widely accepted XACML standard used in access control rule enforcement. The controls within the EUCS scheme, which consist of sets of statements, can be likened to policies. There are several CNLs developed by both academia and industry to articulate such policies. In this context, this whitepaper will explore languages that articulate policies related to data management, and will describe in detail the MEDINA CNL.

The remaining part of this document is structured as follows: Section 2 provides some background information on CNLs in general, and on the CNL that served as a basis for the MEDINA CNL. Section 3 presents a description of the proposed CNL. Section 4 highlights some benefits and drawbacks. Section 5 concludes the document.

(cc) BY-SA

2 Background

Over the past two decades, data protection has become a focal point of discussion across various domains, including critical infrastructures and social networks. Informally, regulations governing data protection, storage, and sharing are typically expressed in natural language. However, bridging the gap between traditional legal contracts that regulate data sharing across different domains and the underlying software architecture that supports them requires a more adaptable approach. CNLs serve as a flexible intermediary for achieving this.

Here are some examples of languages and associated tools designed for secure data management:

- Binder [2] is an open, logic-based security language that encodes security authorizations among components within distributed communication systems.
- The Rodin platform offers an animation and model-checking toolkit for analysing properties of specifications based on the Event-B language, which is capable of expressing security data policies. In a related study [3], the creators of Rodin and Event-B introduced a formalization of data management clauses using Event-B. They employed a model checker to ensure that a system adheres to its associated clauses.
- Another noteworthy contribution comes from [4], which presents a comprehensive framework for articulating highly intricate privacy-related policies, encompassing purposes and obligations.
- The Klaim family of process calculi [5] provides a high-level model for distributed systems, enabling programming and control over resource access and usage.
- Additionally, research in [6] delves into policies that regulate the utilization and replication of information, such as imposing limitations on how often certain information can be used or copied. The analysis tool employed is a static analyser designed for a variant of Klaim.

We will now introduce a language previously developed by members of the MEDINA consortium, known as the *Controlled Natural Language for Data Sharing* Agreement (CNL4DSA) [7], it was created to address several key objectives:

- Reducing Adoption Barriers: CNL4DSA aims to lower the hurdles associated with the adoption of data policies, particularly in the realms of security and privacy.
- Ensuring Formal Policy Mapping: It is designed to ensure that policies can be effectively mapped to formal languages, allowing for automated policy verification [7].

A data sharing agreement essentially constitutes a contract agreed upon by two or more parties, outlining terms and conditions governing data sharing, storage, and utilization. This language, abbreviated as CNL4DSA for brevity, facilitates straightforward yet formal specifications of various categories of privacy policies, as outlined below:

- **Authorizations**: These express permissions for subjects to carry out actions on objects (e.g., data) under specific contextual conditions.
- **Obligations**: Obligations define instances where subjects are required to perform actions on objects under specific contextual conditions.

A central concept within CNL4DSA is the *fragment **f***, which takes the form of a tuple: ***<s, a, o>,*** where ***s*** represents the subject, ***a*** denotes the action, and ***o*** signifies the object. In essence, a fragment conveys that "subject s is performing action a on object o." By introducing "*can*" and "*must*" constructs to the basic fragment, it transforms into an authorization or an obligation, respectively.

CC BY-SA

Fragments are evaluated within specific contexts denoted as ***c*** which are predicates characterizing factors such as user roles, data categories, time, geographical locations, and more. Contexts are assessed as either true or false. Examples of simple contexts include "subject hasRole CSP" or "object hasCategory CloudResource". In order to describe complex policies, contexts are composable. A *composite context C* is defined inductively as follows.

$$C = c \mid \text{C } \textbf{and } \text{C} \mid \text{C } \textbf{or } \text{C} \mid \textbf{not } \text{C}$$

where **and**, **or**, and **not** are Boolean connectors.

The syntax of a composite fragment F_M is described by the following Backus-Naur Form BNF-like syntax:

$$F_M = nil \mid \textbf{mod}\ f \mid F_M; F_M \mid \textit{if } C \textit{ then } F_M$$

where the modality **mod** ranges over {**can**, **must**} and the subscript *M* ranges over {*A*, *O}* where *A* stands for Authorization and *O* stands for Obligation. By changing both the modality and subscript, we can distinguish between two distinct policy categories: authorizations and obligations. FA represents a composite authorization fragment, while FO represents a composite obligation fragment.

Let us now comment on the individual policy constructors:

- *nil* does nothing.
- **mod** *f* is the atomic authorization/obligation fragment that expresses that f = <s,a,o> is allowed/obliged. Its informal meaning is that "*subject s can/must perform action a on object o"*.
- F_M ; F_M is a list of composite fragments. (Subscript *M* takes either only *O* or only *A*).
- *If C then* F_M expresses the logical implication between a composite context *C* and a composite fragment F_M: if *C* holds, then F_M is permitted/obliged (according to the value of *M*).

CNL4DSA employs an operational semantics rooted in a Modal Transition System (MTS), enabling the expression of *permissible* and *mandatory* requirements for CNL4DSA specifications. Modal transition systems are utilized to represent the specifications' behaviour. In its original version [8] MTS is a structure

$$(\mathcal{A}, \mathcal{S}, \rightarrow\Diamond, \rightarrow\boxdot)$$

where $\mathcal{S}$ is a set of specifications, like for example processes in the context of Process Algebras,

$\mathcal{A}$ is the set of actions which specifications may perform, and $\rightarrow\Diamond, \rightarrow\boxdot \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ are the two modal transition relations expressing admissible and necessary requirements to the behaviour of the specifications.

In particular, $S \xrightarrow{a} \Diamond\ S'$ with S, S' $\in \mathcal{S}$ and $a \in \mathcal{A}$ means that it is admissible that the implementation of *S* performs a and then behaves like S'. Dually, $S \xrightarrow{a} \boxdot S'$ represents a transition in which the implementation of *S* is required to perform a and then behaves like *S'*.

This works under the assumption that all the required transitions are admissible transitions.

Figure 1 shows the operational semantics of F_A in terms of a modified label transition system $MTS_{Auth} = (\mathcal{AUT}, \mathcal{F}, \rightarrow\Diamond, \mathcal{C})$. As usual, rules are expressed in terms of a set of premises, possibly empty (above the line) and a conclusion (below the line).

CC BY-SA

Let $\rightarrow_{\Diamond}$ be the smallest subset of $\mathcal{AUTH} \times \mathcal{F} \times \mathcal{AUTH}$, closed under the following rules:

$$(can)\frac{}{can\, f \xrightarrow{f}_{\Diamond} nil} \qquad (if)\frac{C = true \quad F_A \xrightarrow{f}_{\Diamond} F'_A}{if\ C\ then\ F_A \xrightarrow{f}_{\Diamond} F'_A}$$

$$(;)\frac{F1_A \xrightarrow{f}_{\Diamond} F1'_A}{F1_A;F2_A \xrightarrow{f}_{\Diamond} F1'_A;F2_A}$$

Figure 1. Operational Semantics for the composite authorization fragment, where the symmetric rule for (;) is omitted (source: unpublished manuscript, Petrocchi M and Matteucci I.)

MTS_{Auth} deals with authorized transitions only and it also considers the set of contexts because the transitions may depend also on the value of such contexts, see rule (*if)* in Figure 1.

The introduction of $\mathcal{C}$ (= a set of predicates) in a labelled transition system is a standard practice [9]. We observe that the *if* operator implies the binding of variable appearing in the context *C.*

The operational semantics of F_O is expressed in terms of the modal transition system MTS_{Obl} = $(OBL, \mathcal{F}, \rightarrow\boxdot, \mathcal{C})$. The axioms and rules are similar to the ones presented for F_A apart from changing the transition relation, that becomes $\rightarrow\boxdot$.

CC BY-SA

3 MEDINA CNL

In this section, we introduce the MEDINA CNL.

Inspired by the presence of the authorization and obligation modalities in CNL4DSA, presented in Section 2, we further analysed the EUCS draft candidate certification scheme and were able to summarize the requirements, being either technical or organizational [10] in the following general textual formula:

The value assumed by the metric x on the resource of type y can/must be equal/major to/minor to the target value z/must fall in the range of values {z, ...w}.

Paraphrasing part of CNL4DSA, we define a MEDINA *fragment* as a tuple

$$f = \langle rt, m, type(op,tv) \rangle$$

where *rt* is a resource type, *m* is a metric, *type(op,tv)* specifies the type of the metric, the designed target value *tv* for that metric, and the operator *op* which relates the value of the metric to *tv* (e.g., *=*, *>*, *<*, etc.).

A MEDINA fragment says that "the metric *m*, measured on the resource type *rt,* has a specific relation with the value *tv* of type *type*, based on the operator *op"*.

This leads to the following syntax for the MEDINA CNL:

$$F_M = nil \mid \textbf{mod}\ f \mid F_M; F_M$$

where *M* ranges over {*A*, *O*} (Authorization/Obligation) and

- nil does nothing.
- **mod** *f* is the atomic authorization/obligation MEDINA fragment that expresses that *f = <rt, m, type(op,tv)>* is allowed/obliged. Its informal meaning is that *"the metric m, measured on the resource type rt, can/must have a specific relation with the value tv of type type, based on the operator op".*
- F_M ; F_M is a list of composite MEDINA fragments.

mod changes depending on whether one wants to express authorizations or obligations:

can *<rt, m, type(op,tv)>* or ***must*** *<rt, m, type(op,tv)>*

We remind the reader that ***RT*** is the resource type, ***M*** is a metric associated to the EUCS requirement, ***tv*** is a target value, ***op*** is the comparison operator, which indicates how to compare the target value with the value measured on the resource, with respect to metric ***M***; finally, ***type*** indicates the unit of measure of the target value and the measured value.

The language chosen to represent MEDINA's requirements is a simple language. However, we argue that the language is suitable for MEDINA's purpose, which is to create a close-to-standard language for the representation of cloud certification requirements, and which is then made machine readable and input to the assessment tools developed within the project, described in MEDINA Deliverable D3.3 [10].

The language described here was defined as part of the MEDINA research project, and more information about its use and the components surrounding can be found in the MEDINA deliverable D2.5 [11].

CC BY-SA

4 Benefits and drawbacks

Incorporating the MEDINA Controlled Natural Language into cybersecurity certification schemes, particularly within the European Union Cloud Security Certification Scheme (EUCS), introduces significant advantages as well as some challenges. This section delves into these benefits and drawbacks, offering a comprehensive perspective on the implications of utilizing the MEDINA CNL in automated compliance assessments.

One of the most compelling advantages of embracing MEDINA CNL lies in the realm of enhanced automation. By translating natural language controls into a structured, machine-readable format, this CNL streamlines the process of evaluating whether cloud service providers adhere to certification requirements. This heightened level of automation reduces the reliance on human assessors, leading to faster certification procedures and cost savings. Moreover, the MEDINA CNL significantly improves the accuracy of compliance assessments. The potential for ambiguity inherent in natural language controls is effectively mitigated, resulting in more precise evaluations of compliance. Complex or vaguely defined requirements are no longer susceptible to misinterpretation, reducing the chances of both false positives and false negatives in compliance assessments. Standardization and consistency are additional benefits associated with MEDINA CNL, which enforces a uniform framework for expressing certification requirements, ensuring that all controls are structured in a consistent manner. This standardization simplifies the task of comparing and assessing compliance across various Cloud Service Providers and fosters the creation of unambiguous audit trails. As a result, certification processes become more efficient, which is particularly advantageous for Cloud Service Providers seeking rapid certification to gain a competitive edge.

Nevertheless, transitioning from traditional natural language-based compliance assessments to CNL-based assessments may present a learning curve for assessors and organizations. The need for training and familiarization with the CNL could potentially lead to an initial slowdown in certification processes. Moreover, the complexity of expressing certain requirements represents another challenge. While the CNL simplifies the representation of most controls, highly intricate or context-specific requirements may prove difficult to accurately convey within the language. This limitation may require additional efforts to refine CNL constructs for complex scenarios. In addition, there is the possibility of oversimplification. In some cases, the CNL may simplify requirements to the point of omitting nuanced aspects of certification controls. Finally, achieving full interoperability between different CNLs and certification schemes may necessitate significant coordination and standardization efforts. The absence of such interoperability could potentially lead to challenges for organizations working across multiple certification schemes employing different CNLs.

(cc) BY-SA

5 Conclusions

In this whitepaper, we have explained the central role of the MEDINA Controlled Natural Language in the context of cybersecurity certification schemes, with a particular focus on the European Union Cloud Security Certification Scheme (EUCS). This activity has resulted in a multitude of benefits, even though some associated challenges inherent in integrating MEDINA CNL into certification processes have been revealed.

The adoption of MEDINA CNL represents a significant leap forward in automating compliance assessments. By translating natural language controls into a structured, machine-readable format, this CNL streamlines the certification process, leading to faster assessments, cost savings, and reduced human error. The potential for ambiguity is significantly mitigated, enhancing the precision of compliance evaluations.

However, this transition to CNL-based assessments presents also challenges. In conclusion, the adoption of the MEDINA CNL offers substantial advantages in automating compliance assessments within cybersecurity certification schemes. While the journey may involve some hurdles, the benefits of increased efficiency, accuracy, and transparency outweigh the challenges.

(cc) BY-SA

6 References

[1] ENISA, "EUCS – Cloud Services Scheme," [Online]. Available: https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme . [Accessed Sept. 2023].

[2] M. Abadi, "Logic in access control," in *LICS*, 2003.

[3] A. Arenas, B. Aziz, J. Bicarregui and M. Wilson, "An Event-B approach to data sharing agreements," in *IFM, LNCS 6396*, 2010.

[4] Q. Ni and e. al, "Privacy-aware Role-based Access Control," *ACM Transactions on Information and System Security,* 2010.

[5] R. De Nicola, G. Ferrari and R. Pugliese, "Programming Access Control: The KLAIM Experience," in *CONCUR 2000. LNCS, vol. 1877*, 2020.

[6] R. Hansen, F. Nielson, H. Nielson and C. Probst, "Static validation of licence conformance policies," in *ARES*, 2008.

[7] Ilaria Matteucci, Marinella Petrocchi, Marco Luca Sbodio: "CNL4DSA: a controlled natural language for data sharing agreements," in SAC 2010: 616-620

[8] K. Larsen and B. Thomsen, "A modal process logic," in *Third Annual Symposium on Logic in Computer Science*, 1988.

[9] J. Bergstra, A. Ponse and S. Smolka, Handbook of Process Algebra, Elsevier, 2011.

[10] MEDINA Consortium, "D3.3 - Tools and techniques for the management of trustworthy evidence v-3," 2023.

[11] MEDINA Consortium, "D2.5 – Specification of the Cloud Security Certification Language v-3," 2023.

(cc) BY-SA