D4.1

Architectural Specification of CASTOR Continuum-Wide Trust Assessment Framework

Project number:	101167904
Project acronym:	**CASTOR**
Project title:	Continuum of Trust: Increased Path Agility and Trustworthy Device and Service Provisioning
Project Start Date:	1st October, 2024
Duration:	36 months
Programme:	HORIZON-CL3-2023-CS-01
Deliverable Type:	Report
Reference Number:	HORIZON-CL3-2021-CS-01-101167904/ D4.1 / v1.0
Workpackage:	WP4
Due Date:	30st December, 2025
Actual Submission Date:	9th February, 2026
Responsible Organisation:	COLLINS
Editor:	Stelios Basayiannis, Michael McElligott
Dissemination Level:	Public
Revision:	1.0
Abstract:	This deliverable documents initial work on the Trust Assessment Framework, capturing core terminologies and guidance towards the trust management lifecycle. It provides state-of-the-art analysis on different trust modelling approaches, contributing to the motivation of employing Subjective Logic as the foundational mechanism behind TAF. Secondly, it presents a preliminary set of trust relationships and a high-level description of the overall TAF within the routing plane. It emphasizes the need for continuous risk analysis to build trust models for trust decisions. Finally, it presents a concrete formulation of the problem as an optimization task and outlines potential approaches to address it, enabling the eventual translation of trust into actionable traffic engineering controls.
Keywords:	Trust Assessment, Subjective Logic, Optimization, Trusted Path Routing

Funded by EU's Horizon Europe programme under Grant Agreement number 101167904 (CASTOR). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

Funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee.

Copyright Notice

Project Funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable	R*	
	Dissemination Level	
PU	Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)	X
SEN	Sensitive, limited under the conditions of the Grant Agreement	
Classified R-UE/ EU-R	EU RESTRICTED under the Commission Decision No2015/ 444.	
Classified C-UE/ EU-C	EU CONFIDENTIAL under the Commission Decision No2015/ 444	
Classified S-UE/ EU-S	EU SECRET under the Commission Decision No2015/ 444	

* R: Document, report (excluding the periodic and final reports)
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc.
DMP: Data management plan
ETHICS: Deliverables related to ethics issues
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Editor

Stelios Basayiannis, Michael McElligott (COLLINS)

Contributors (ordered according to beneficiary numbers)

Nikos Fotos, Thanassis Giannetsos (UBITECH)
Iasonas Sakellariou, Stelios Kazazis, Symeon Tsintzos (QUBITECH)
Michael McElligott, Stelios Basayiannis (COLLINS)
Alexandros Fakis, Kostas Maliatsos (FERON)
Anuj Pathania, Andy Pimentel (UvA)
Jamie Pont, Budi Arief, Theo Dimitrakos (UKENT)

Disclaimer

Executive Summary

In modern, heterogeneous network environments, the routing plane remains a critical yet opaque layer where unmapped trust dependencies introduce systemic vulnerabilities. For highly sensitive workloads requiring robust security assurances, the current lack of such trust dependency scoping prevents organizations from accurately identifying and mitigating emerging threats. To bridge this gap and embed trust as an intrinsic component of network traffic engineering, two fundamental challenges must be addressed.

First, organizations require the capability **to systematically measure trustworthiness across the architecture; scaling from low-level device telemetry to high-level insights that inform service-wide decisions**. This requires moving beyond initial trust establishment to a model of continuous runtime characterization throughout the operational lifecycle of the network. Second, this real-time visibility creates the opportunity to treat trust not merely as a static prerequisite, but as a dynamic decision point for path selection. Solving this second challenge involves **mapping trust insights into actionable traffic engineering policies that satisfy performance requirements while optimizing for trust in a multi-objective framework**.

While Deliverable D2.1 [5] introduces the overarching CASTOR architecture and its requirements for orchestrator-driven trusted path routing, Deliverable D3.1 [7] provides the technical blueprint for the device-level Trusted Computing Base. Together, these documents set the stage for secure evidence collection, providing the necessary data layer for the systematic trust evaluations performed within the CASTOR framework. Building upon these foundations, this deliverable specifies the initial CASTOR Trust Assessment Framework (TAF). It defines the critical terminology and trust evaluation modalities required to establish a unified semantic baseline and operational language for all subsequent WP4 activities.

Following the terminological specification, this deliverable evaluates the state of the art in trust frameworks to identify critical gaps in current assessment methodologies. This analysis informs a detailed set of requirements for trust evaluation in the routing plane, providing the rationale for why a framework based on Subjective Logic offers superior evaluative capabilities over alternative and well-established decision logics. Subsequently, this deliverable defines the core trust modelling principles necessary for a multi-layered and federated evaluation strategy that spans from local, in-router trust properties to trust relationships established both between forwarding-plane elements and between routers and the centralized Orchestration Layer.

Based on this, this document presents a high-level architecture for a Trust Assessment Framework (TAF) instance and details its position within the broader CASTOR ecosystem. Beyond the runtime trust engineering process, which enables the systematic quantification of the Actual Trustworthiness Level (ATL), the deliverable defines the challenges of deriving a robust trust decision methodology by introducing a risk-aware Required Trustworthiness Level (RTL). The document evaluates current approaches to modelling and deriving RTL thresholds and identifies the core challenges of identifying a robust RTL methodology; specifically one capable of capturing complex network interdependencies and the potential for cascading attacks within the forwarding plane.

To translate these evaluations into actionable policies, the deliverable formulates the co-enforcement of network and trust requirements in traffic engineering as a multi-objective optimization problem. A targeted analysis of exact and heuristic algorithms is provided, establishing the algorithmic basis for the CASTOR Optimization Engine.

Ultimately, this deliverable forms the basis for the development of the CASTOR Trust Assessment Framework and the CASTOR Optimization Engine. By framing core challenges through self-contained Engineering Stories, it sets the scene for the specification of the functional requirements and driving factors that will shape the development of all WP4 artifacts in subsequent phases of the project.

Contents

List of Figures

List of Tables

Versioning and contribution history

Version	Date	Author	Notes
v0.1	27.10.2025	Nikos Fotos (UBITECH)	Vocabulary definition, and first input on Chapter 7 for the overall Trust Assessment Framework
v0.2	10.11.2025	Jamie Pont, Theo Dimitrakos (UKENT)	First draft of State-of-the-art analysis in probabilistic logics
v0.3	17.11.2025	Jamie Pont (UKENT), Alexandros Fakis (FERON)	Input on Subjective Logic primitives, and Trust Relationships (Chapter 6). First Input on Engineering Stories
v0.4	24.11.2025	Jamie Pont (UKENT), Alexandros Fakis (FERON)	Input on Subjective Logic primitives, and Trust Relationships (Chapter 6), Draft input on RTL in Chapter 8
v0.5	5.12.2025	Iasonas Sakellariou, Symeon Tsintzos (QUBITECH)	State-of-the-art analysis on optimization techniques in Chapter 9
v0.6	12.12.2026	Jamie Pont (UKENT), Alexandros Fakis (FERON), Anuj Pathania (UvA)	Document trust assessment modalities in Chapter 3, Elaborate on the possible equations for RTL derivation in Chapter 8, Review optimization SotA (Chapter 8)
v0.7	19.12.2026	Jamie Pont (UKENT), Alexandros Fakis (FERON), Iasonas Sakellariou, Symeon Tsintzos (QUBITECH)	Case Study analysis in Chapter 4, Final adjustments in Risk Assessment Engineering Stories and Chapter 8, Final updates to Chapter 9 on optimization
v0.8	15.1.2026	Nikos Fotos (UBITECH), Anuj Pathania (UvA)	Final updates to Engineering Stories (Chapter 10) and Chapter 7.
v0.9.0	23.1.2026	Nikos Fotos, Thanassis Giannetsos (UBITECH)	Intro/Conclusions and final polishing of deliverable.
v0.9.1	30.1.2026	Nikos Fotos, Thanassis Giannetsos (UBITECH)	Revised Executive Summary
v1.0	9.2.2026	Daphne Galani (UBITECH)	Final Review & Submission

Chapter 1

Introduction

1.1 Demystifying Dynamic Trust Characterization in Network Traffic Engineering Process

Emerging requirements for robust service provisioning increasingly demand the distribution of computational resources from centralized cloud infrastructures to far-edge environments. Evaluating the trustworthiness of the application service lifecycle therefore becomes a critical challenge in achieving high levels of assurance, particularly for highly sensitive workloads. Within this edge-to-cloud paradigm, an integral component is the trust evaluation of the transport network that spans and interconnects the entire compute continuum. In this context, the systematic and continuous assessment of the dynamic trust relationships within the forwarding plane emerges as one of the principal challenges for next-generation Connected Collaborative Computing Networks ("3C Networks") [18].

CASTOR is the first of its kind to address this challenge in a holistic manner. To this end, CASTOR identifies two key objectives that together enable end-to-end trust characterization: **measuring trust** and **co-enforcing trust and network requirements**. The first objective focuses on the systematic modelling and evaluation of trust relationships within the highly volatile and rapidly evolving routing plane. This effort is intrinsically linked to the development of a comprehensive trust assessment framework capable of deriving critical trust insights that characterize the security posture of an end-to-end network path and its participating administrative domains. The second objective addresses the challenge of translating these trust insights into meaningful and actionable traffic engineering decisions. Specifically, CASTOR investigates how this translation can be formulated as an advanced optimization problem, thereby enabling the recommendation of network paths that simultaneously satisfy multiple network- and trust-related requirements and can be enforced within the forwarding plane.

In WP4, we focus on achieving these goals by investigating how trust is modelled, assessed, and managed within the overarching CASTOR framework. The core outcomes of this activity are (i) the development of an overarching trust assessment framework that will enable the evaluation of trust properties across various aspects of the routing plane and (ii) the optimization engine that will shape the traffic engineering insights (e.g., recommended network paths) that can be then used to enforce a new set of routing policies that can accommodate various sets of network- and trust- requirements as envisioned in the context of the use cases.

Overall, in alignment with the Zero Trust paradigm, CASTOR does not assume any inherent trustworthiness in the forwarding plane. Instead, trust evaluations are grounded in the secure collection of trustworthiness evidence obtained from the Trust Sources available at each network element. As illustrated in Section 5.4, the presence of multiple evidence sources—often incomplete or potentially conflicting—leads us to employ Subjective Logic as the primary probabilistic framework to guide trust evaluations and subsequent trust-aware recommendations. Chapter 6 captures how it is possible the different trust relationships

pertaining to the forwarding plane, covering both trust dependencies at the management plane (e.g., between an orchestration service and a router) but also at the forwarding plane (e.g., between two adjacent routers).

As elaborated below, this document outlines an initial set of requirements and key concepts that will inform the functional specification of the aforementioned artifacts in subsequent work package activities. It presents a high-level overview of the CASTOR Trust Assessment Framework and introduces the core concepts of trust modelling based on continuous risk analysis, as well as trust-aware optimization in the traffic engineering domain.

1.2 Relation with other WPs and Deliverables

Deliverable D4.1 serves as a strategic guide for the development of all artifacts comprising the CASTOR Trust Assessment Framework (TAF), facilitating the integration of trust characterizations into the determination of near-optimal network paths. Then these insights can inform the design of routing policies that simultaneously satisfy both network (i.e., performance) and trust objectives. In the following sections, we outline the core requirements that will shape the CASTOR TAF and the CASTOR Optimization Engine, enabling continuous assessment of network trustworthiness: from in-router evaluations to comprehensive, topology-wide trust assertions.

The CASTOR WP4 activities follow a clear roadmap from initial requirements to successive releases (as shown in Figure 1.1. This document (D4.1) presents crucial information with respect to the specification of categories of atomic and composite trust propositions, the main types of trust relationships at device, path, and domain-levels, and the definition of the risk-aware Required and Actual Trustworthiness Levels (RTL and ATL respectively) that dictate the trust characterization of the CASTOR TAF. In addition, it formulates the problem of deriving traffic engineering policies as a multi-objective optimization problem. This formulation enables the identification of the primary exact and heuristic algorithms that the Optimization Engine will explore to derive recommended traffic engineering policies. Building on these inputs, the first release (D4.2) operationalizes the WP4 roadmap by delivering an initial version of the trust models tailored to traffic engineering evaluations, as well as the overall CASTOR Trust Assessment Framework and its interactions with the available Trust Sources. In addition, it introduces the first version of the Optimization Engine, including preliminary results on the accuracy and robustness of the different optimization strategies it supports.

The subsequent (and final) release, documented in D4.3, extends this work by specifying trust relationships that include router-to-router evaluations within the forwarding plane, in line with IETF Trusted Path Routing. It also extends the dynamic trust models to capture cross-domain relationships. This, in turn, enables the evaluation of both the CASTOR Trust Assessment Framework and the Optimization Engine in cross-domain scenarios, as envisioned in the use cases, thereby completing the framework's end-to-end assessment and policy derivation capabilities.

WP2 provides the foundational requirements and the high-level architectural vision for the Risk Assessment Engine, the CASTOR Trust Assessment Framework (TAF) and the Optimization Engine, which directly inform the scope and design choices of WP4. In particular, WP2 defines the system-level constraints, operational assumptions, and functional expectations that WP4 translates into concrete trust models, optimization formulations, and architectural building blocks. This linkage ensures that the technical developments in WP4 remain aligned with the overall CASTOR objectives and system architecture.

WP3 contributes the requirements and conceptual architecture of the CASTOR Trusted Computing Base (TCB) and identifies the trust sources that can be leveraged within the framework in order to securely collect the relevant trustworthiness evidence for the ATL derivation. These inputs are essential for WP4, as they determine the types of trust propositions that can be evaluated based on verifiable claims and evidence provided by the trust sources. WP4 builds upon this input to define atomic and composite trust

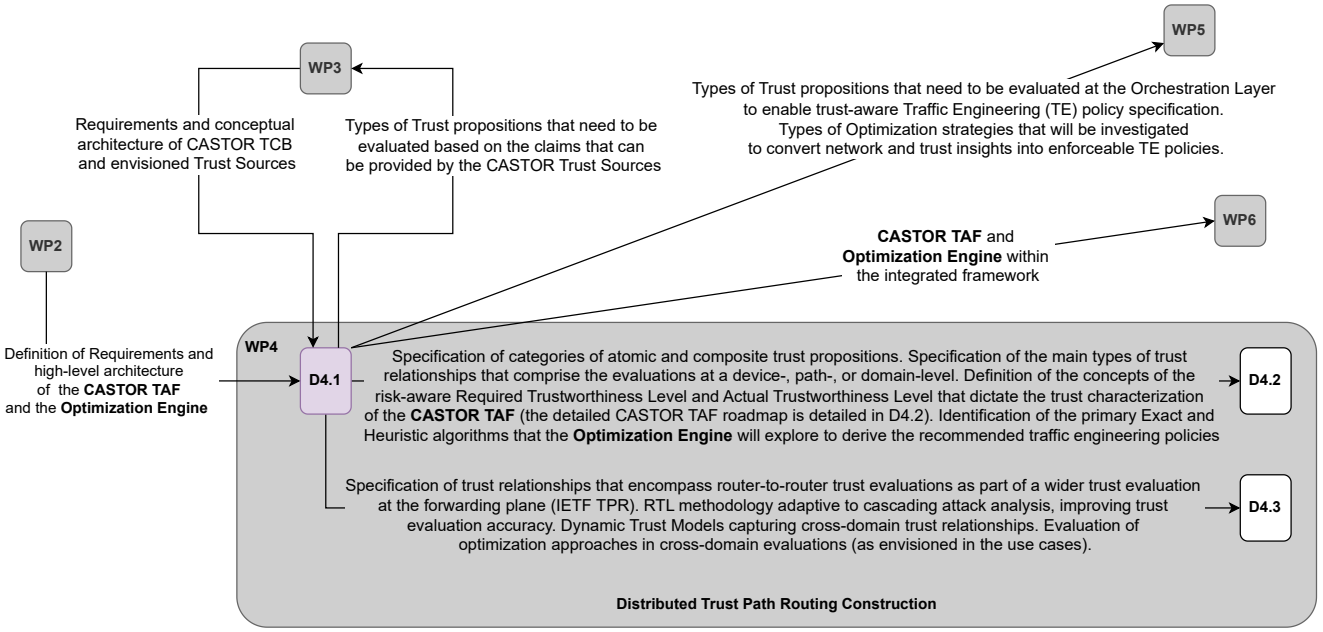

Figure 1.1: Relation of D4.1 with other WPs and Deliverables

propositions and to structure trust relationships that are compatible with the capabilities and guarantees offered by the CASTOR TCB.

In the context of WP5, D4.1 provides an initial analysis on the type of trust propositions and trust-aware abstractions that must be evaluated at the orchestration layer to enable trust-aware Traffic Engineering (TE) policy specification. Based on the trust relationships and evaluation mechanisms developed as well as the optimization strategies that combine network-level metrics and trust assessments, WP5 is able to derive accurate TE policies and enforce them through its different probes in the network. This interaction ensures that orchestration decisions are grounded in formally defined and continuously assessed trust information.

WP4 provides WP6 with the architectural definition of the CASTOR TAF and the Optimization Engine as integrated components within the overall framework. This connection enables the practical exploitation of trust-aware routing and traffic engineering decisions, ensuring consistency between trust assessment, optimization logic, and domain-level network operations.

In brief, this deliverable represents an initial milestone for CASTOR toward the establishment of an overarching trust assessment framework. Grounded in the principles of Zero Trust — i.e., where no implicit trust is assumed and all decisions are derived from continuously collected trustworthiness evidence — and supported by Subjective Logic as a formal reasoning framework for handling uncertainty and conflicting information, this work consolidates the core architectural foundations of CASTOR. These foundations enable the systematic characterization of trust relationships beyond individual entities, extending toward path-level trust evaluation across federated domains, and directly supporting trust-aware traffic engineering decisions within complex and dynamic network environments.

1.3 Deliverable Structure

This document begins by establishing the conceptual foundations of the CASTOR Trust Assessment Framework (TAF).Chapter 2 introduces and clarifies the terminology associated with the core components and operational capabilities of the framework, followed by a presentation of the different trust assessment modalities envisioned in CASTOR (see Chapter 3). These framework variants are outlined in terms of their targeted functions, setting the stage for a structured and incremental realization of trust assessment

mechanisms. Before delving into the technical details, Chapter 4 establishes a solid conceptual foundation by formally defining Trust and Trustworthiness, elaborating on their nuances, and analysing the shortcomings of traditional trust verification approaches and how CASTOR aims to address them.

Building upon these conceptual foundations, Chapter 5 then surveys existing approaches to trust management with a particular focus on probabilistic logics that are able to cope with uncertain and contradicting evidence. Relevant state-of-the-art research is examined, highlighting strengths, limitations, and open challenges especially in the context of network traffic engineering. Various trust reasoning and decision logic approaches are discussed, culminating in a rationale for adopting Subjective Logic as the primary reasoning framework, due to its ability to explicitly model uncertainty, incomplete knowledge, and conflicting evidence commonly encountered in dynamic network environments.

The following two chapters address the architectural and methodological realization of the CASTOR TAF. Chapter 6 identifies the trust relationships underpinning trust-aware traffic engineering policies at both in-router and network levels, while Chapter 7 outlines the core functionalities of a TAF instance, paving the way toward the functional specifications of standalone and federated trust assessment frameworks in D4.2. This organization clearly distinguishes between local (in-router) and global (orchestration-level) trust evaluations.

While previous chapters focus on the evidence-based evaluation of the Actual Trustworthiness Level, Chapter 8 presents the need for a risk-aware Required Trustworthiness Level methodology that will allow a relying entity to make informed decisions on where a network entity can be considered trusted for a given context and scope. The chapter continues by introducing approaches on how a thorough and continuous risk assessment process can contribute to the derivation of accurate RTL values that can be used to form the overall Trust Policy that will guide and dictate the Trust Engineering process throughout the operational lifecycle of the network topology.

Chapter 9 introduces the problem of trust-aware traffic engineering as a multi-objective optimization problem. Starting from the establishment of a common vocabulary this chapter formulates the optimization problem by integrating network and trust attributes into a unified problem that can provide important recommendations for deriving the appropriate traffic engineering policies in the forwarding plane. A concise state-of-the-art analysis of exact and heuristic optimization methodologies is presented, covering both classical and emerging approaches applicable to multi-objective optimization in the routing domain. Collectively, this chapter provides the foundation for specifying the functional requirements of the CASTOR optimization engine and directly informs its design choices and implementation roadmap in D4.2.

Finally, Chapter 10 presents the engineering stories that capture the main challenges and requirements related to the overall Risk and Trust Engineering process as well as the Optimization Engine. The Engineering Stories shed light into critical aspects of each artefact, unlocking their detailed functional specification in deliverable D4.2. Chapter 11 discusses the final remarks and conclusions of this deliverable.

Chapter 2

Trust Management Terms And Definitions

This chapter defines the terminology used within the context of trust management and the wider Trust Assessment Framework (TAF) of CASTOR itself. It is intended that this chapter can be used as a single point-of-reference for the reader to better understand the various concepts and terminologies of trust assessment discussed throughout the project.

ATL (Actual Trustworthiness Level) An Actual Trustworthiness Level (ATL) is the quantifiable result of a specific evaluation of an atomic or composite trust proposition within the scope specified by the Trustworthiness Level Expression Engine (TLEE). The resultant ATL can be interpreted as the extent to which a given node, link, path or data can be considered trustworthy based on the available evidence.

ATO (Atomic Trust Opinion) An Atomic Trust Opinion (ATO) is the specific subjective logic opinion formed by the Trust Source Manager (TSM) in relation to a Trust Source (TS) (based on available trustworthiness evidence), in the context of a specific trust relationship.

Atomic Trust Proposition An atomic trust proposition is a specific, atomic (i.e. cannot be further broken down into sub-propositions) logic statement about an entity for whom trust is to be assessed. An Atomic Trust Opinion (ATO) is subsequently formed in relation to a given atomic trust proposition.

Composite Trust Proposition A composite trust proposition is the logical combination of two or more atomic trust propositions, allowing for reasoning about the trustworthiness of higher-level and more abstract concepts. Composite trust propositions can be broken down into their atomic counterparts to allow for the aggregation of individual Atomic Trust Opinions (ATOs) into a singular opinion capturing the trustworthiness of the entire composite proposition.

Data-Centric Trust Data-centric trust refers to a trust assessment in the context of a data-centric trust relationship. In such a scenario, the trustor is a node (such as a router), and the trustee is the data itself.

Discounting Discounting is a component of Subjective Logic (SL) that accommodates the modulation of trust opinions with respect to the perceived opinion of the reporting entity. For example, a Global TAF can discount an opinion received from a Local TAF agent based on its own opinion of the Local TAF's ability to generate trust opinions in a reliable and trustworthy manner.

Federated Trust Assessment Federated Trust Assessment is a core contribution of the CASTOR framework. This modality supports two core functionalities. Firstly, a TAF is able to make trust assessments based on evidence and opinions it has received from *neighbouring nodes*. This is a step up from the standalone modality, where a Local TAF can only assess integrity based on locally

collected evidence. The benefit here is that through the use of Subjective Logic (SL), multiple trust opinions and evidence can be *discounted* and *fused* to form a single consolidated opinion representing trustworthiness. Secondly, building on from this point, a TAF is able to make trust assessments on entities to which it is *indirectly* linked. This process, known as *referral trust*, allows the Global TAF to form a more complete picture of the overall network by establishing opinions on newly onboarded entities (to which it does not yet have a direct link) by relying on (and discounting) opinions from Local TAFs to which the new node is directly linked.

Fusion Fusion is a component of Subjective Logic (SL) that facilitates the combining of multiple trust opinions that relate to a singular phenomenon of interest into a single trust opinion that appropriately weights all constituent opinions. This is useful when, for example, combining the opinions of multiple Local TAF agents at the Global TAF level to evaluate a trust proposition. Various fusion operators exist, such as *Cumulative Fusion*, *Epistemic Fusion* and *Consensus Fusion*.

Link-Level Trust Link-level trust refers to the extent to which a link (i.e. network channel) between two nodes can be considered trustworthy. Evidence in this context may relate to both the nodes present in the given link, as well as evidence directly relating to the link itself, such as available bandwidth.

Logical Connectives Logical connectives are the operators used to aggregate operands (which, in the context of CASTOR, refer to trust propositions). Examples of connectives include *conjunction* (AND) and *disjunction* (OR), and are used by the TAF to aggregate trust propositions into higher abstractions such as link and path-level trust propositions. Importantly, these composite trust propositions can subsequently be decomposed by the Trustworthiness Level Expression Engine (TLEE) using a *decomposition function*, enabling the fusion of individual Actual Trustworthiness Levels (ALTs) through Subjective Logic (SL).

Node-Centric Trust Node-centric trust refers to a trust assessment in the context of a node-centric trust relationship. In such a scenario, both the trustor and trustee are a node (such as a router).

Node-Level Trust Node-level trust is the concept of trustworthiness at the level of an individual node, such as a specific router. In other words, this term is in relation to the extent at which a single node can be considered trustworthy, and is based on node-level propositions such as secure boot attestations.

Path-Level Trust Path-level trust refers to the extent at which an entire path throughout the network, from point of ingress to point of egress, can be considered trustworthy. It is formed through the logical composition of link and path-level trust propositions relating to entities participating within the link.

Referral Trust Referral trust is an indirect trust relationship between two trust objects. The trustor can evaluate the trustworthiness of the trustee by evaluating the opinions of one or more intermediate nodes that form the indirect link between the two nodes in question. It is important that referral trust opinions are discounted by the evaluating entity, based on the perceived credibility of the intermediary nodes.

RTL (Required Trust Level) The Required Trustworthiness Level (RTL) is the level of trustworthiness that a given application considers acceptable in order to consider the node or data in question to be trustworthy, such that it can be relied upon at runtime.

Standalone Trust Assessment Stadalone Trust Assessment refers to the idea of a single Local TAF agent forming opinions on trust opinions by evaluating evidence that it has collected *locally*. This modality is limited to evaluating integrity in order to reduce bottlenecks, and creates the foundation upon which Federated Trust Assessment is built.

TA (Trust Assessment) Trust Assessment (TA) is the collective concept of trustworthiness level evaluation and decision making performed within the Trust Assessment Framework (TAF).

TAF (Trust Assessment Framework) The CASTOR Trust Assessment Framework (TAF) is a software framework that is able to evaluate trust sources and trustworthiness evidence in the context of a given trust model to derive an Actual Trust Level (ATL). This ATL can subsequently be used by the Trust Decision Engine (TDE) in conjunction with an associated Required Trust Level (RTL) in order to calculate actionable trust decisions to facilitate network operation.

TDE (Trust Decision Engine) The Trust Decision Engine (TDE) performs the final steps of a trustworthiness evaluation before the output of a trust assessment is delivered. The output can either be in the form of an Actual Trust Level (ATL) (as calculated by the Trustworthiness Level Expression Engine (TLEE)) or a trust decision.

TLEE (Trustworthiness Level Expression Engine The Trustworthiness Level Expression Engine (TLEE) is a core component of the TAF that is tasked with calculating the trustworthiness of a given atomic or composite trust proposition within the context of the specified trust model. The TLEE operates on subjective logic to form an Actual Trust Level (ATL) based on Atomic Trust Opinions (ATO)s from the Trust Source Manager (TSM).

TM (Trust Model) A Trust Model (TM) is a graph-based model built upon a system model representing all components and data needed to perform a certain function. Components either create, transmit, process, relay or receive the data used as input to the function. Vertices in a TM correspond to Trust Objects (TOs), and edges correspond to trust relationships between pairs of TOs. The TM also includes a list of Trust Sources (TSs) used to quantify trust relationships by providing Atomic Trust Opinions (ATOs). The TM is the main input to the Trust Model Manager (TMM) and Trustworthiness Level Expression Engine (TLEE). As trust is a directional relationship between two TOs and is always in relation to a concrete property or scope, then the TM can encompass multiple trust relationships between the same two TOs depending on different properties of the trust relationship or its scope.

TMM (Trust Model Manager) The Trust Model Manager (TMM) stores the Trust Models (TMs) and makes them available to components of the TAF that require them, such as the Trustworthiness Level Expression Engine (TLEE).

Trust Trust represents a decision (or disposition) by a trustor to place, or withhold, trust to a specific trustee. If a trustor decides to trust a given trustee, the trustor believes that, with high confidence, the trustee will fulfil the trustor's expectations. Trust is a property of the trustor.

Trust Object A trust object represents an entity that either assesses trust (of another entity or data), or for which trust is to be assessed, and is represented in a trust model as a vertex.

Trust Policy The Trust Policy, defined by the network operator, is a set of guarantees that ensures the infrastructure layer adheres to the Security Service Level Agreements (SSLAs). It merges the Required Trustworthiness Level (RTL), evidence types to be collected, and relevant Trust Models (TMs) that ultimately inform the trust assessment process. Different trust policies are specified based on the type of node (i.e. router) and operational phase (i.e. onboarding or runtime), and they are distributed via the CASTOR Distributed Ledger Technology (DLT) to ensure appropriate configuration during the runtime phase.

Trust Relationship A trust relationship represents a directional relationship between two trust objects, where the "trustor" is assessing the trustworthiness of the "trustee". A trust relationship is always directional and in relation to a specific trust property and scope, and is represented in a trust model as an edge.

Trustee A trustee is an entity in the trust model that aims to fulfil the expectation of another entity (the trustor).

Trustor A trustor is an entity in the trust model that has a certain requirement, and an expectation that this requirement will be fulfilled by another entity (the trustee).

Trustworthiness Given the knowledge that a trustor trusts a trustee, it can be said that the trustor believes the trustee to have the property of trustworthiness. Trustworthiness is a property of the trustee.

TS (Trust Source) A trust source (TS) manages trustworthiness evidence inside the TAF. It can quantify the trustworthiness of a trustee based on this evidence in the form of an atomic trust opinion (ATO) when requested by the Trust Assessment Framework (TAF).

TSM (Trust Source Manager) The Trust Source Manager (TSM) handles all available Trust Sources (TSs) inside the Trust Assessment Framework (TAF). It can also establish new TSs dynamically through a plugin interface.

Chapter 3

Trust Assessment Modalities

The CASTOR architecture aims to redefine trust assessment by defining trust as a continuous and quantifiable property. To accommodate trust assessment within the vast and heterogeneous Compute Continuum (CC), CASTOR implements two trust assessment modalities, namely *standalone trust assessment* and *federated trust assessment*.

In short, the standalone trust assessment modality is concerned with local trust assessment, performed exclusively at the level of the Local TAF, in relation to evidence collected by the Local TAF agent itself (note that the sources of evidence may be external to the Local TAF, however) and *only* in relation to integrity. No other Local TAF, nor the Global TAF itself, will be used in conjunction with the Local TAF performing a trust evaluation in this modality. This approach allows for faster and less safety-critical trust evaluations to be completed without the overhead and potential bottlenecks of relying on the wider system. The federated trust assessment modality, on the other hand, extends upon the functionality of the standalone modality by incorporating additional Local TAFs, and the Global TAF, to make more accurate trust assessments in relation to higher-level phenomenon and additional trust properties (such as availability) that involve more than one TAF, as well as allowing for indirect trust relationships between participating entities (known as referral trust).

3.1 Standalone Trust Assessment

The standalone trust assessment modality is entirely focused on the operation of the Local TAF agent, operating at the level of a node in the network such as a router. The primary focus of standalone trust assessment is the self-assessment and trustworthiness quantification of integrity-related trust propositions, all performed locally to the TAF in question. There is one exception to this rule, in the case where the Global TAF operates on direct evidence from a router, that has not been assessed by a Local TAF. This is discussed in Section 3.1.3, and visualised in Section 6.1.1.

3.1.1 Optimisations

The standalone trust assessment modality is realised through the necessity of extreme computational efficiency in the domain of highly resource-constrainted network devices such as routers. To ensure fast and robust trust assessment during runtime without imposing perceptible delay in the wider system, the Local TAF implements several optimisations that minimise its scope and overhead without compromising its ability to perform local trust assessments. For example, trust assessment in this modality is restricted to integrity-based trust assessments, and avoids heavy use of complex subjective logic operators such as fusion and discounting. These processes are generally reserved for the federated trust assessment modality, discussed in Section 3.2. This architectural separation reduces the computational overhead on

the Local TAF and affords low-latency input to the wider federated trust assessment model and overall network environment.

This trust assessment modality solely operates on static (e.g. configuration) and dynamic (e.g. runtime behaviour) properties of the node. The Local TAF's purpose is the continuous assessment of trustworthiness relating to the specific device on which it is running, and its execution is within a Trusted Execution Environment (TEE).

The Local TAF exclusively evaluates on *evidence collected locally*, forming atomic trust propositions that it can subsequently evaluate in the form of trust opinions. In particular, the internal architecture of the device is treated as a single agent system model, reducing the complexity of the related trust model that must be stored and used at the local level. A simple key-value database maps each trust relationship between its corresponding atomic trust proposition and analyst node (i.e. the Local TAF agent itself). In short, this decision reduces overloading on resource-constrained devices (such as routers) through the use of less complex trust models and by minimising the requirement of computationally expensive subjective logic operators.

As previously stated, trustworthiness is only assessed in this modality in the context of the integrity trust property. Several other properties of trust are considered, namely confidentiality, availability and robustness, but these properties are assessed in conjunction with integrity in the federated trust assessment modality. This decision again reduces computational overhead at the local level and prevents the Local TAF agent from becoming a bottleneck to the wider system.

3.1.2 Core Operations

As discussed in the previous sections, the core functionality within the standalone trust assessment modality is designed with efficiency and reduced computational overhead in mind given the resource-constrained environment in which it must run. The scope of operation focuses on transforming locally collected evidence into quantifiable trust opinions concerning the given node's integrity. This process generally involves three main steps: evidence collection, opinion quantification, and local evaluation against the enforced trust policy.

Evidence collection. The Local TAF communicates with its own local trust sources (which may be in the form of e.g. attestation reports) that provide the evidence for which the trust assessment is to be based. This evidence typically pertains to information such as configuration and behavioural runtime traces. The Trust Policy defines the exact types of evidence to be collected for a given trust assessment and ensures that the focus is exclusively on evidence relevant to the required guarantees.

Primary trust sources within this context include the core attestation components relating to platform state, such as secure boot and configuration measurements, as well as the Finite State Machine (FSM) agent (running within the router's Trust Network Device Extensions (TNDEs)), which reports evidence relating to any observed misbehaviour of the reporting node (for example protocol violations or abnormal runtime operation).

Opinion quantification. The Local TAF quantifies trust opinions strictly for atomic trust propositions based directly on the locally collected evidence. Crucially, these trust propositions cannot be broken down into more granular sub-propositions, and map directly what can be observed. An example of such a proposition could be that "Secure Boot for this router stands", evaluating to true or false.

It is based on this supporting evidence that an atomic trust opinion is formed regarding the specific trust proposition. Overall, this quantification process results in an Actual Trustworthiness Level (ATL) that can be compared to the Required Trustworthiness Level (RTL) specified by the trust policy, or forwarded to the Global TAF in the case of the federated trust assessment modality.

Local evaluation. This step involves the evaluation of locally-derived ATLs against the RTL constraints

enforced via the Trust Policy. This comparison allows the Local TAF to derive an actionable trust decision about the trustworthiness of the router itself in regards to its integrity.

3.1.3 Global TAF Standalone Trust Assessment

Up to this point, the standalone trust assessment modality has only been discussed within the context of simpler, local trust assessments performed entirely at the node level. However, as previously stated, there is an exception to this behaviour. This is shown in the case where a Global TAF is acting as the direct assessor for some device-level trust property beyond that of integrity, evaluating raw evidence (such as CPU utlisation data) securely provided through the Trust Network Device Interface Security Protocol (TNDI-SP) channel (an example of which is illustrated in Section 6.1.4).

This allows the handling of device-level properties that a Local TAF may not be equipped to evaluate, and circumvents the need for the wider system to wait for the Local TAF to perform a trust assessment in time-critical scenarios. The Global TAF will, however, have to discount the evidence received based on its opinion of the TNDI-SP channel to form an accurate trust opinion. The discounting process is discussed in more detail in Section 3.2 and Chapter 6. It is important to note that in this scenario, all evidence and opinion quantification is performed at the level of the Global TAF rather than the Local TAF, establishing direct trust relationships between the Global and Local TAF despite the fact that the trust assessment was not performed locally. This still falls within the remit of the standalone TAF, as only a single TAF (in this case, the Global TAF) was involved in the evaluation process, without sharing (or receiving) information with other TAFs in the network, and is visualised in Section 6.1.4.

3.2 Federated Trust Assessment

As discussed in the previous section, CASTOR also implements a federated trust assessment modality facilitating more comprehensive trust assessments that make use of additional sources of evidence, multiple TAFs (i.e. an arbitrary number of Local TAFs and the Global TAF) and encompass additional properties of trust such as availability and robustness. Put simply, the federated trust assessment modality expands upon the capabilities of the standalone modality by allowing for the exchange of information between individual TAFs, incorporating subjective logic in order to fuse and discount multiple trust opinions into a single opinion that accurately represents the trustworthiness of a proposition.

In the standalone modality, we focus primarily on the scenario in which a given Local TAF performs the entire trust assessment process *locally*. In other words, evidence was collected and quantified, trust opinions were derived, and a resulting evaluation (i.e. formation of a trust decision) was performed entirely locally to the node running the TAF. In the federated modality, however, most of the processing is performed on the Global TAF after having received information from participating nodes in the trust assessment process. This distinction introduces the concept of *uncertainty*, i.e. an inherent need to modulate received information (opinions and evidence) based on the credibility of the source node.

Subjective logic is used extensively to handle this uncertainty. This functionality allows a Global TAF to *fuse* multiple opinions received concerning a trust proposition into a single, verifiable opinion whose input opinions have been appropriately *discounted* based on the Global TAF's prior opinion on the reporting node itself. The choice of both an appropriate fusion and discounting operator is non-trivial and depends on factors such as the quality of received evidence, as well as the desired treatment of overlap and/or contradictions in evidence. Examples of these scenarios are illustrated in Sections 6.1.2, 6.1.3, 6.1.5, 6.1.6 and 6.1.7. A case study that illustrates the challenges of federated trust assessment is illustrated in Section 4.3. In addition, the processes of fusion and discounting are presented in more detail in Section 5.4.

Chapter 4

General Concepts of Trust and Trustworthiness

This chapter formally introduces and defines the concepts of Trust and Trustworthiness that are briefly detailed in Chapter 2. Section 4.1 defines trust and trustworthiness in the context of trustors and trustees, as well as expands on their nuances. Section 4.2 identifies challenges faced in more traditional trust verification systems and details how CASTOR aims to address these challenges.

4.1 Trust, Trustworthiness and Other Related Terminology

The core concepts of trust and trustworthiness directly relate to (and are properties of) a *trustor* and a *trustee*. A trustor is an entity in the network, such as a router, that has a certain requirement and expectation that this requirement will be fulfilled by some other entity. For example, an expectation may be that a neighbouring node will share information in a timely manner with full integrity. In contrast, the trustee is the entity that aims to fulfil the expectation of the trustor.

As a more concrete example, consider the scenario within the context of the Trusted Path Routing paradigm, where two routers (Router A and B) aim to establish a secure link between them. In this scenario, Router A is the trustor, and may expect Router B (the trustee) to forward packets confidentially and with integrity, all within an acceptable time-frame (i.e. with availability and low latency).

Now that the concepts of trustor and trustee have been established, we delve further into *trust* and *trustworthiness*, two fundamental aspects of Trusted Path Routing that are integral to the CASTOR Trust Assessment Framework (TAF).

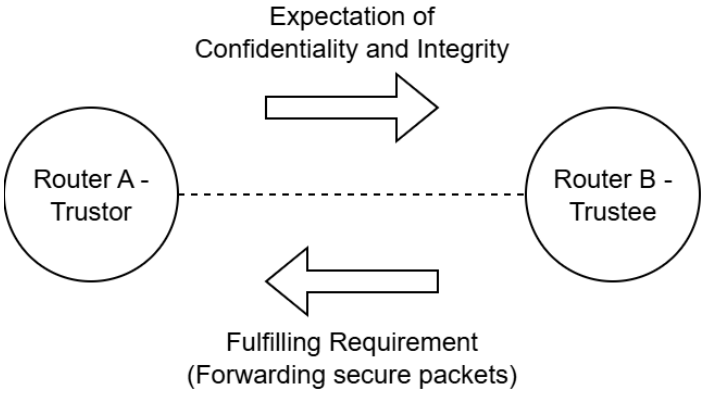

Figure 4.1: A bi-directional trust relationship between two entities. Router A, the trustor, expects Router B, the trustee, to forward packets with confidentiality and integrity.

Trustworthiness. Trustworthiness is defined as *the likelihood of a trustee to fulfil a trustor's expectations in a given context*.

Trust. Trust is defined as *a decision (or disposition) by a trustor to place, or withhold, trust to a specific trustee*. If a trustor decides to trust a trustee, the trustor believes that – with high confidence – the trustee will fulfil the trustor's expectations. Alternatively, given the knowledge that a trustor trusts a trustee, it can be said that "the trustor trusts the trustee" is equivalent to the fact that "the trustor believes the trustee to have the property of trustworthiness (in the given context)".

Whereas trust is a *property of the trustor*, trustworthiness *characterises the trustee*. Trustworthiness itself quantifies the trustee's capacity in fulfilling the expectations of the trustor. The trustor must appropriately evaluate the level of trustworthiness of a trustee such that it can be balanced with the trustor's expectations to ensure trust is truly warranted.

Furthermore, the degree to which a trustee can be considered trustworthy is based on both *their ability to meet the expectations of the trustor* and *the extent to which their capabilities are aligned with the trustor's goal*. Trustworthiness is confined to a specific *context*, i.e. a trustee is *only* expected to fulfil the goal of a trustor given a set of restrictions and/or circumstances. In other words, a trustee's trustworthiness is evaluated in relation to the specific trustworthiness assessment process, and the outcome does not influence any other trust assessment in which the trustee may be participating. A specific trustworthiness assessment is *isolated*, and as an example, if a trustee fails to meet the requirements of a trustor in one particular instance, that is not to say that another trustor cannot deem the same trustee to be trustworthy in another context. After all, trust verification is *subjective*.

The expectation of a trustor can relate to *data* or to the *behaviour of the trustee* itself. For example, there may be an expectation of data integrity (i.e. the data has not been modified during transit), but there may also be an expectation relating to the functionality of the trustee itself (i.e. the entity, such as a router, must be forwarding information confidentially, consistently and within a specified time-frame).

The collection and quantification of evidence provides the trustor with the ability to assess the likelihood with which a trustee is able to fulfil its request. More generally, this process known as *trust verification*, directly influences the trust decision-making process. Evidence can exist in several forms, for example:

- Proof of the trustee's past behaviour, ideally in the same (or a similar) context to that of the current task,
- Independent assessments made by other entities regarding the trustee's ability to achieve the current task,
- Information on regulatory constraints that may allow or prohibit the trustee from completing the current task.

4.2 General Challenges of Trust Assessment

In the more traditional paradigm, which incorporates a one-time and binary trust model, trust is granted or denied at the point of ingress. This perimeter-based approach provides minimal to no consideration of the dynamic nature of complex network environments [29]. In the following section, we detail some of the general limitations and challenges in the domain of trust assessment, as well as how CASTOR aims to address these challenges.

4.2.1 Continuous, Non-Binary Trust Verification

The challenge of moving beyond a one-time binary trust model is at the heart of the design of the CASTOR TAF. In more traditional approaches, an entity is simply labelled as “trusted” or “untrusted” during onboarding. There are several limitations with this design. Perhaps most notably, an entity only has to convince the network operator that they are trustworthy during the onboarding phase. Although this may be sufficient for entities joining the network in good faith, it does not consider the long-term behaviour of the entity.

For multiple reasons, an entity that has previously enrolled in the network can begin acting in ways that would be considered suspicious or even unacceptable. The most obvious case is that of a determined attacker who, after gaining access to the network, begins to act maliciously, similar to the concept of an advanced persistent threat (APT) [2]. However, even entities acting in good faith may unintetionally exhibit unwanted behaviour on the network. Hardware, both in terms of participating devices and network infrastructure, degrades over time. This can lead to reduced bandwidth, increased latency, and concerns about availability, along with increasing the potential attack surface by inadvertently introducing vulnerabilities to the overall system. All of the above impact an entity’s trustworthiness by introducing more uncertainty into the system.

CASTOR seeks to address these limitations by continually assessing trust for all participating nodes, links and paths over the lifetime of the network. A TAF agent must be able to recognise such a degradation in operational performance and adjust its trust opinion appropriately, for example by increasing its level of disbelief in a given trust proposition. Recall that belief, disbelief and uncertainty are *belief mass* values and must sum to one. Therefore, increasing disbelief implies a reduction in belief, uncertainty or both. This is exacerbated in the federated trust assessment modality, where a Local TAF must also maintain an up-to-date opinion on its neighbouring nodes so that received opinions and evidence can be appropriately discounted, discussed in more detail in Section 4.3.

Furthermore, the CASTOR architecture explicitly redefines trust from a binary concept to a continuous and quantifiable process. The trustworthiness of entities participating in the network is evaluated along a scale that is automatically and continuously updated in response to changing network conditions and entity behaviour, more accurately reflecting the degree to which they can be trusted. This is realised through the incorporation of subjective logic, allowing for the reasoning of evidence in the presence of uncertainty to calculate *trust opinions*, rather than static one-time labels. Through the use of fusion and discounting operators (discussed in more detail in Section 5.4), opinions derived by multiple TAFs can be effectively aggregated, resulting in a more accurate assessment of a node’s behaviour over time based on input from multiple participants in the network.

4.2.2 Evidence Quantification and Subjectivity

The complexity of trust assessment is compounded by the need to handle and evaluate diverse sources of evidence that can be incomplete and often contradictory. Although evidence itself is objective, the process of trust verification is subjective – different TAFs may interpret the same pieces of evidence differently based on their own trust policies. This ultimately means that two trustors, given the same piece of evidence, may produce different trust verification results (even to the extent that one trustor deems the evidence as sufficient to grant trust, whereas another does not).

In addition, not all sources of evidence are equal, and may provide information in varying formats. This makes the evidence quantification process complex, as despite the heterogeneity of evidence sources, a TAF must be able to ultimately aggregate all collected evidence and form a single trust opinion. As such, evidence quantification functions that convert raw data and evidence into formal trust metrics must be used to appropriately map the available evidence to trust opinions. Furthermore, the quality and nature

of available evidence (for example how much it contradicts or overlaps) directly informs which subjective logic fusion operator to prioritise such that resulting trust opinions are not influenced incorrectly.

4.3 Challenges of Federated Trust Assessment

One of the primary contributions of the CASTOR project is the federated trust assessment modality that allows for multiple TAF agents to interact and exchange information to perform accurate trustworthiness assessments. This federated trust model consists of a single Global TAF and multiple Local TAFs that can independently quantify evidence and form trust opinions that are ultimately aggregated at the global level.

However, this modality introduces additional complexity. An opinion formed by one TAF cannot be assumed to be perfectly accurate every time. For reasons previously discussed, nodes can degrade in performance or even act maliciously, and therefore relying on their output verbatim may result in inaccuracies in the trust verification process. To mitigate this threat, the Global TAF must *discount* any opinions and/or evidence received from other entities participating in the network. Discounting is a component of subjective logic that allows an opinion to be modulated based on the perceived opinion of the reporting entity. For example, if the Global TAF receives an opinion from a node that is known to be lacking in terms of trustworthiness, then the opinion can be heavily discounted to lessen its impact. Conversely, opinions received from highly-trusted nodes can be granted more weight and have a greater influence in the overall resulting trust decision.

One practical application of this process is in the case where an onboarding node doesn't have a path established with the Global TAF yet. By relying on opinions received from neighbouring nodes of the newly onboarded node, the Global TAF can discount and *fuse* them form an indirect trust relationship with the newly onboarded node. Fusion is another component of subjective logic that allows independent opinions concerning the same phenomenon to be effectively "averaged" into a single representative opinion. As with the choice of discounting operator, the choice of fusion operator also plays an integral role in ensuring accurate trust assessments. This scenario is discussed in more detail in Section 6.1.7 and 6.1.8. The concepts of SL discounting and fusion are presented in detail in Section 5.4.

4.3.1 Case Study: A Simple Illustration of Federated Trust Assessment Flow

The following section presents a simple example of a network environment in which CASTOR is assessing trust, as a vehicle to demonstrate the delicate nuances of federated trust assessment discussed in Section 4.3. In Figure 4.2, we examine the trust engineering process taking place at the Global TAF level on a subnetwork containing six separate routers, each with its own Local TAF agent. The figure represents the overall flow, starting with the risk assessment and RTL derivation (discussed in detail in Chapter 8), and moving to the overall trust assessment process (further discussed in Section 7.3.

The Risk Assessment Asset topology assigns a *risk score* to each available asset (e.g. router) in the topology. This can be based on prior knowledge of known threats and vulnerabilities in each participating asset on a per-router basis, or can be derived based on interdependencies and the potential for cascading attacks between assets. The latter is highly dependent upon the asset's positioning within the topology. For example, u_5 has four directly connected routers with a larger potential for compromise and an increased risk score. Of course, the number of connections is not the only contributing factor; u_2 also has four connections but only a moderate risk. This is because the resultant risk score is derived based on a combination of both individual and interdependency risk. In other words, in the example we consider that the vendor-specific u_2 asset has less critical vulnerabilities than u_5. Similarly, u_3 and u_4 have severe risk despite fewer connections, due to the higher individual risk. This step allows for the derivation of the relevant trust policies that can subsequently be implemented in the Global TAF and Local TAF

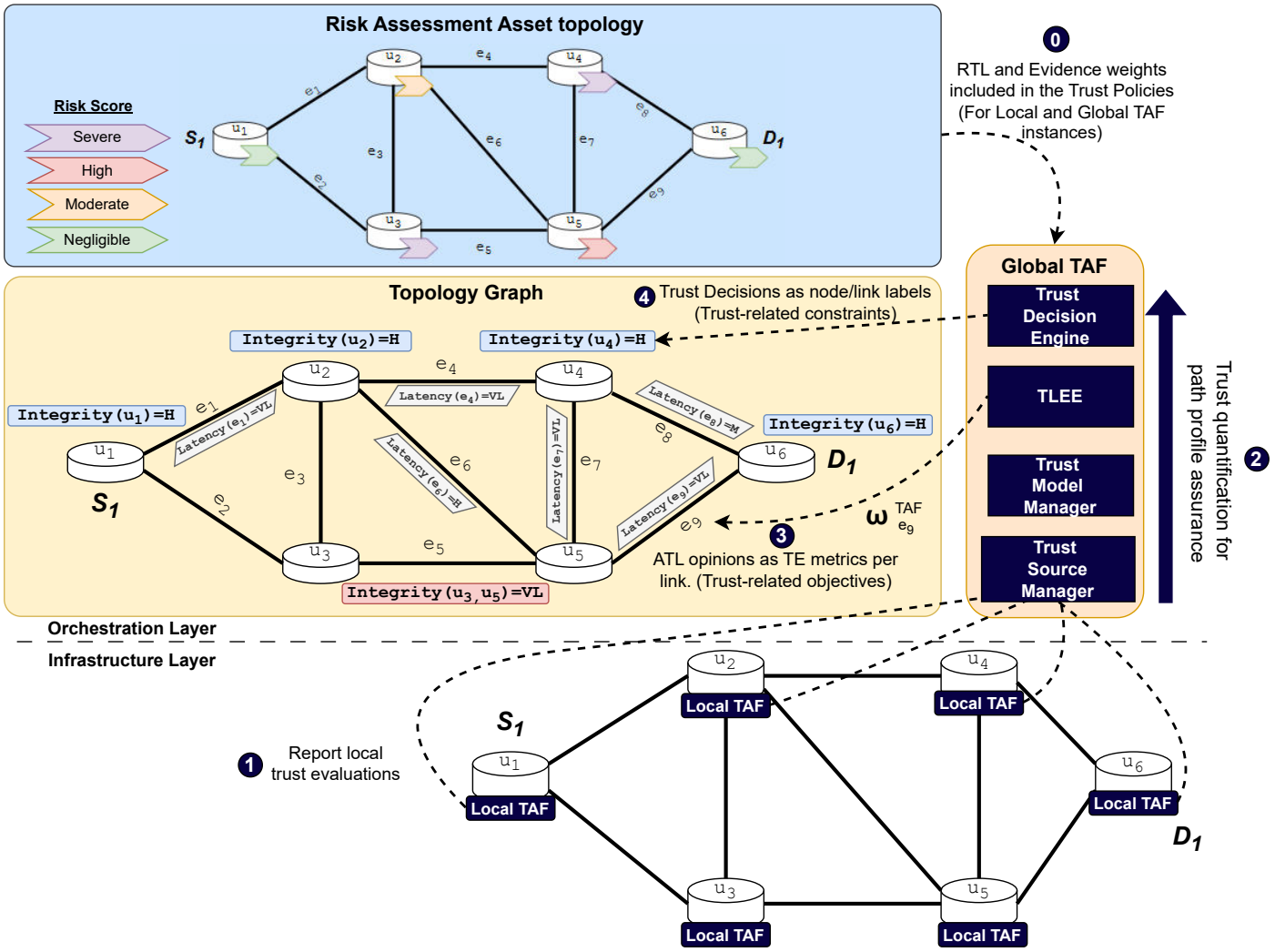

Figure 4.2: Running example - CASTOR TAF federation information flow

agents. Overall, this risk analysis heavily contributes to the derivation of the Trust Policy that each Local TAF agent and the Global TAF shall run during runtime in order to accurately measure and evaluate the trustworthiness for a desired context (i.e., trust property) and scope (i.e., evaluation at a node-, link-, or path-level).

Based on the enforced Trust Policies across the infrastructure layer, the Local TAF agents are able to interact with the in-router trust sources (all part of the CASTOR TNDE presented in D3.1 [7]), in order to securely collect fresh trustworthiness evidence related to the target trust evaluations. In this step, depending on the enforced Trust Policy, the Trust Sources may share their information directly to the Global TAF or the Local TAF agents may run some trust calculations locally and then report their outcomes to the Global TAF. As further detailed in Section 5.4, these reports may take the form of binomial (i.e. a state space of two outcomes, such as “True” or “False”) or multinomial (i.e. a state space larger than two, such as “Low”, “Medium” or “High”) opinions. At this stage, the Global TAF can quantify trust to ensure that the requirements of the domain operator are met. Details on the high-level architecture of a TAF instantiation are presented in Chapter 7.

The Trust Source Manager constitutes an integral part in the TAF’s evidence quantification process, as it serves as the interface with supported Trust Sources in order to process incoming trustworthiness evidence. In CASTOR, we focus on two primary in-router Trust Sources as illustrated in D3.1 [7]: the Attestation Source that provides runtime guarantees on the configuration and software correctness of critical network functions, and the Finite State Machine Source that moniotores the runtime operation of the router and probides evidence on its behavioural correctness. Of course, as part of the TAF federation

modality (explained in Chapter 3), the Trust Source Manager shall also process trust evaluations coming from other TAF agents as illustrated in this example too. All these different types of trustworthiness evidence allow a TAF instance to form normalised trust opinions and store them in the corresponding trust relationships at the Trust Model Manager. These opinions can be, then, passed to the TLEE which calculates the final ATLs for the end-to-end trust evaluations. These ATL values correspond to the final trust score that the Global TAF assigns to different trust entities: network elements, links, paths or even entire domains. As part of the service assurance mechanisms described in D5.1 [6], the final ATLs produced by the Global TAF shall be mapped to the trust requirements of each service with respect to its workload traffic. At the same time, the Global TAF invokes its Trust Decision Engine in order to compare the derived ATL values with the corresponding RTL values that are provided in the Trust Policy. This comparison allows the derivation of a trust decision with respect to the trustworthiness of an entity for a given context. Consequently, the output of the Global TAF is twofold: the ATL value which corresponds to the evidence-based estimation of the Global TAF on the trustworthiness of an entity, and ii) the trust characterization/decision of an entity based on the associated RTL value. Both values are used to populate the trust-related profile of each entity in the Topology Graph (as shown in Steps 3 and 4 of Figure 4.2. These trust evaluations represent the Global TAF's perspective on the trustworthiness of the underlying network topology, incorporating any available evidence, including Local TAF opinions. In addition to trust-related information, the Topology Graph maintains a corresponding characterization of each node and link in the network, as provided by the Network Service Orchestrator (The network labels in the Topology Graph are intentionally shown in a smaller font, as this topic is discussed in D5.1 [6]).

The Topology Graph construction constitutes the core knowledge that allows us to address the challenge of trust-aware traffic engineering provisioning. As discussed in Chapter 9, CASTOR treats this challenge as a multi-objective optimization problem as the information within the Topology Graph enables the identification of traffic engineering strategies that are able to satisfy both network- and trust-related objectives and constraints. An example inspired from the topology graph of Figure 4.2 could be that a service provider requires from a network service orchestrator a forwarding path from S1 to D1 where we achieve the maximum integrity while the path does not include a link with latency exceeding the tier "Very Low" (VL). Apart from the trust-related labels in the Topology Graph as a result of the trust decision outcome, the Global TAF assigns ATL opinions on the graph. As further explained in the formulation of the general problem in Section 9.2, the Optimization Engine may treat the ATLs as a metric that can be used for maximizing a required trust-related objective. Consequently, depending on the number of trust-related objectives and constraints for multiple trust propositions, the optimisation engine can opt to solve the problem either through the use of hard restraints (such as "avoid paths with low integrity"), through statements based on ATLs (such as "select path with maximum link availability"), or a combination of both. In principle, decision of whether to use the trust-related labels or scores from the Topology Graph is heavily dependent on the path profile requirements (as mentioned in the established Service Level Agreement (SLA).

As a final remark, it is worth noting that all trust evaluations are defined by the trust models, and their semantics are uniform across different routing elements, independent of device vendor. For instance, an ATL value of 0.5 must be interpreted consistently, regardless of the routing element to which it is assigned. While this is straightforward within a single administrative domain, it introduces additional challenges when characterizing trust for end-to-end paths spanning multiple domains. These cross-domain considerations, particularly regarding the interpretability of trust scores across different trust models, will be addressed in the second release of the CASTOR Trust Assessment Framework.

Chapter 5

State-of-the-art in Trust Assessment Methodologies

This chapter presents an analysis of the state-of-the-art in Trust Assessment. We start by outlining various decision-making mechanisms as well as their strengths and weaknesses, before justifying CASTOR's convergence on Subjective Logic (SL) as the chosen mechanism. A detailed overview of SL is then provided, covering concepts such as trust opinions, binomial and multinomial opinions, discounting and fusion.

5.1 Requirements

CASTOR focuses on the dynamic evaluation of trust in complex and heterogeneous network environments and as such demands certain requirements that can be used both in determining the most appropriate decision logic mechanism and overall solution design.

Trust Assessment in Complex Networks

At the forefront, CASTOR operators in highly dynamic network environments with varying and often unpredictable participants, where drastic changes in operational performance can change from one moment to the next. CASTOR must be able to handle dynamic network conditions, whether they are caused by malicious threat actors, user error (such as misconfiguration), or infrastructure degradation. Furthermore, each agent in the topology must be capable of contributing towards the overall assessment of trust, and as such, their various inputs must be carefully weighted and appropriately aggregated. To accurately reflect trust, the system must be able to operate not only on fresh evidence but also take previous evidence into consideration, as well as provide a metric for determining the *freshness* of information (the value of which influences evidence weighting). The framework should operate across multiple domains, each with bespoke network operators, path profile requirements and trust models, and be able to assess trustworthiness at multiple levels of granularity, namely data-level, node-level, link-level, path-level and domain-level.

Trust Assessment in the Presence of Uncertainty

Trust modelling inherently comes with a level of uncertainty. Evidence and information received are often incomplete, disjointed and contradictory. This *epistemic* uncertainty leads to situations in which determining an absolute truth is complex and nuanced. CASTOR must be able to quantify and reason about the perceived level of uncertainty when performing trust assessments, using it to more accurately evaluate trustworthiness and influence actionable trust decisions, rather than "averaging out" or simply discarding any notion of uncertainty and/or ignorance. For example, if two routers are sharing highly contradictory opinions to the Global TAF concerning a newly onboarded router, the Global TAF must be

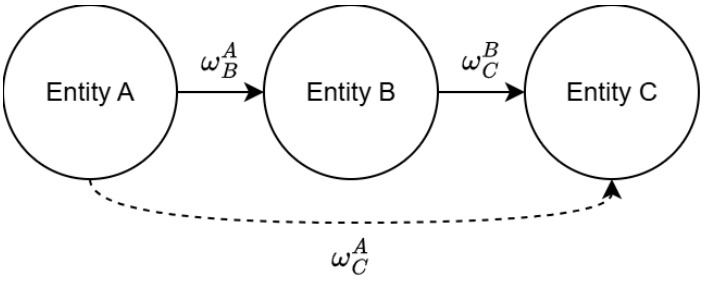

Figure 5.1: A simplified illustration of trust transitivity. Entity A does not have a direct relationship with Entity C, but can receive a referral from Entity B (to which it does have a direct trust relationship). The conceptual "transitive" relationship is shown as a dotted arrow.

able to systematically take both opinions into consideration, carefully fusing them (for example by using an epistemic fusion operator that places priority on uncertainty to represent the disagreement) into a single, accurate trust opinion. This resulting opinion may have high levels of uncertainty, signaling to the network orchestrator that not enough information is available to safely route traffic through that particular segment.

Probabilistic Trust Propositions

CASTOR must support expressive trust propositions that can be combined with logical connectives to build higher levels of assertions representing the trust properties that are in scope for the project, such as integrity and availability. In contrast, composite trust propositions must be able to be broken down into their smallest constituent parts (referred to as atomic trust propositions) that can be evaluated independently in order to accurately calculate trustworthiness levels. Supported trust propositions must not only include binomial state spaces (i.e. propositions that evaluate to True or False), but also multinomial state spaces. Multinomial trust propositions allow trust decisions to operate on more granular results (such as whether integrity is evaluated as "low", "medium", "high" or "severe").

Dynamic Support of Varying and Multi-Source Evidence

In CASTOR, we adhere to the theory of Zero Trust. In this modality, trust is never assumed and must always be measured through evidence-based theory. As previously explained, evidence can be contradictory and can be either static (e.g. attestation reports) or probabilistic (e.g. runtime behaviour). To cater for this nature of evidence, probabilistic logic is needed to build trust relationships and systematically measure trust. In the context of CASTOR, vRouters could be reconfigured to accommodate services with stronger guarantees. For example, the risk assessment process dictates the number and type of security controls that need to be enforced during runtime for trust characterisation. The risk assessment process runs continuously, and topology or cyber-threat intelligence updates could lead to dynamic reconsideration of the collected evidence, leading to updates in the Trust Policy.

Furthermore, in CASTOR Deliverable 3.1, a detailed threat model is captured that encompasses different security properties and router behaviours of a TNDI. Each threat is tied to a set of raw traces (such as syscalls) that are captured during runtime. Eventually, the TAF must be able to accommodate multiple trust sources in order to facilitate an accurate and overarching trust evaluation of the routing plane.

Federated Trust Assessment, Fusion and Discounting

CASTOR operates in highly complex, multi-agent network environments where there is often no guaranty that any two given entities have direct links between them. As such, it is crucial that the framework supports a federated trust assessment modality; that is, one in which multiple TAF agents can work together to provide evidence and trust opinions on their neighbouring nodes to help form the overarching picture of trust throughout the topology, end-to-end. This demands several sub-requirements that must be supported by the chosen decision logic mechanism.

Firstly, as defined in Chapter 4, a trust relationship exists between a trustor and a trustee. However,

and particularly in complex network environments, there is no guaranty that this relationship is direct (i.e. that the evidence is passed directly from the trustee entity to the trustor entity). Therefore, the trust assessment framework must accommodate indirect (transitive) trust between entities, allowing evidence and opinions to pass through a chain of trust relationships. In the simplest of scenarios, entity A may not have a direct relationship with entity C for which it is assessing trust. However, if entity B has a direct trust relationship with entity A and entity C, entity B can pass its opinion on entity C to entity A directly, enabling entity A to form a trust opinion on entity C. In this situation, trust discounting is crucial. This scenario is illustrated in Figure 5.1.

Second, there must be a mechanism that supports the systematic aggregation of multiple trust opinions that concern the same trust proposition. This concept, known as *fusion*, is critical in ensuring that no information from any participating entity is unfairly discarded without consideration. Fusion allows for an arbitrary number of opinions to be fused into a new single opinion that effectively encapsulates the opinion of a new conceptual agent that represents all participating entities. Careful selection of an appropriate fusion operator is necessary, largely dependent on how agreement and disagreement should be handled (for example by prioritising uncertainty or by weighting opinions based on source trustworthiness).

Finally, an opinion that an agent receives from another agent should not be taken at face-value. Instead, the opinion should be modulated with respect to the credibility of the reporting agent. This process, formally referred to as *discounting*, is vital to mitigate the impact of low-quality or malicious referrals. For example, if a node that is widely regarded as untrustworthy provides a referral, it can be heavily discounted to ensure that it does not over-influence the overall trust evaluation. In a sense, this process implicitly rewards “good behaviour” as information received from known trustworthy nodes will usually have more of an influence in the trust assessment process.

Node vs. Data-Centric Trust

The trust assessment framework must be able to evaluate the trustworthiness of data as well as nodes. Trustworthiness with respect to data can depend not only on the trustworthiness of the data itself but also on the node that produced those data. Therefore, the semantics of both types of trust should be understood, and it should be possible to combine them where necessary. As a result, in the event that data is received from a node for which trust cannot be evaluated, it should still be possible to evaluate the trustworthiness of the data. This decoupling ensures that data-centric trust can be evaluated without relying on the supplying node.

Safe Operation under Time-Critical Restraints

The entire trust assessment process must be performed under strict time restraints and be fit-for-purpose in real-time. This is exemplified in the various use cases specified in CASTOR Deliverable 2.1, for example, the need to monitor airspace in Urban Air Mobility (UAM) environments and to facilitate rapid response for First Responder Mobile Units. In these high-risk scenarios, it is vital that the low latency and high integrity of communication is ensured in real-time, because it is often the case that lives may be at risk and a system failure could be catastrophic.

5.2 Existing Decision Logic Mechanisms

The challenge of assessing trust in the context of Trusted Path Routing has been previously investigated through various methodologies. This section explores some of these mechanisms of decision logic, namely Probabilistic Logic, Fuzzy Logic, Bayesian Probability, Dempster-Shafer Theory and Subjective Logic.

5.2.1 Probabilistic Logic

Probabilistic logic is a mechanism that begins to bridge the gap between traditional logical structures that are simply evaluated as true or false, with the probabilistic modelling of uncertainty [35]. Probabilistic logic assigns likelihoods, along a scale of $[0, 1]$, to propositions, allowing a system to cater for noise and uncertainty with mathematical foundation. This distinction lends itself towards the modelling of trust in network environments that, by their very nature, are noisy, dynamic and often working on incomplete evidence.

However, probabilistic logic is not equipped to deal with the complexities of a real-life setting in this context. For example, it is unclear how to systematically aggregate heterogeneous sources of evidence that may be both partial and unreliable. In addition, probabilistic logic evaluation can become computationally expensive in complex systems such as the network environments in which CASTOR operates.

5.2.2 Fuzzy Logic

Fuzzy logic is an alternative decision logic mechanism that is designed to model the gradation of real-world phenomena that are not accurately conveyed by traditional Boolean logic. Rather than imposing a strict threshold, fuzzy logic assigns propositions "degrees of truth", on a scale of $[0, 1]$, better encapsulating the inherent vagueness in concepts such as "high latency" or "moderate integrity". After all, different individuals may interpret these claims in different ways. This functionality improves applicability in observations that are continuous and have inherent uncertainty, and allows a system to capture more subtle shifts in behaviour that a traditional Boolean classification may be overly or under-sensitive towards. Furthermore, fuzzy logic can be more computationally lightweight than probabilistic logic, allowing for more practical application [36].

In practice, fuzzy logic uses membership functions to encapsulate the extent to which an input belongs to a certain category (i.e. where on the scale it belongs). The degrees of truth are combined through fuzzy rules and aggregated to produce the final decision. However, it relies on manually-tuned rule sets to infer category membership. These rules are subjective and difficult to establish. Traditionally, they are also static and do not automatically adapt as the network environment evolves over time, leading to outdated rules and potentially inaccurate trust assessments in the long term. However, advances in fuzzy logic, such as adaptive fuzzy systems, incorporate learning procedures to update rule sets and membership functions on the fly [40].

5.2.3 Bayesian Probability

Bayesian probability is capable of reasoning in the presence of uncertainty by treating probability as *belief*. Belief is updated in response to received evidence using Bayes' Theorem, resulting in probabilities that account for prior knowledge as well as observed phenomenon [21]. A model defines probability distributions over variables, and a likelihood function is implemented that facilitates evidence quantification. During inference, beliefs are dynamically updated as new observations are made. As such, Bayesian probability is better-suited towards dynamic and volatile environments with ever-changing variables, such as a complex network environment. In addition, it supports the fusion of information and provides a probabilistic representation of confidence that can inform an overall decision making process. Bayes' Theorem is shown below:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$P(A)$ represents the prior probability of event A being true, and $P(A|B)$ is the probability of A being true given the probability of B being true. As new evidence arrives, probabilities concerning existing events are updated and can subsequently be used as prior probabilities for subsequent calculations.

However, Bayesian models rely on manually-defined prior beliefs that represent confidence before new evidence is received. This process is fundamentally important to the accuracy of the overall output, but can be challenging to correctly configure. On top of this, Bayesian models can quickly become computationally expensive limiting their ability to scale in large and dynamic environments. Furthermore, the likelihood function must be defined properly to accurately reflect the environment in which the model is running. These drawbacks are particularly evident in an adversarial setting, where an attacker may attempt to manipulate the model in such a way as to mask their behaviour.

5.2.4 Dempster-Shafer Theory

Dempster-Shafer Theory (DST), developed by Dempster and Shafer [14, 44], extends the modelling of uncertainty by assigning belief to a set of possible outcomes. It allows for the merging of independent sets of beliefs in relation to evidence collected from independent agents. The degree of belief is referred to as *mass*, represented as a belief function rather than a Bayesian probability distribution. Belief and plausibility functions then capture the minimum and maximum support that evidence provides for a proposition. Evidence obtained from multiple independent sources can also be fused using Dempster's Rule of Combination, which accounts for overlap and disagreement amongst sources. DST is used in a range of fields that require decision making with fusion capabilities, such as pattern recognition and sensor fusion [46].

The main components of DST are the basic belief assignment (mass), belief and plausibility. Let $\Theta = \{\theta_i | i = 1, ..., n\}$ denote the frame of discernment (the set of all possible hypotheses) for a given problem. This frame contains n mutually exclusive and exhaustive hypotheses that describe the possible outcomes of a state variable v. The power set 2^Θ is the collection of all subsets of Θ.

A basic belief assignment (BBA) m on Θ (the belief mass function) maps each subset in 2^Θ to a value in $[0, 1]$, subject to:

$$m(\theta) = 0 \text{ and } \sum_{X \subseteq \Theta} m(X) = 1$$

Each value $m(X)$ denotes the level of support that the given evidence lends to the proposition X. When $m(\theta) = 0$, the mass function is referred to as *normal*. Any subset $X \subseteq \Theta$ with $m(X) > 0$ called as a *focal element*.

From a BBA m, it is possible to derive the associated belief and plausibility measures, Bel and Pl, defined for any $X \subseteq \Theta$ by:

$$\text{Bel}(X) = \sum_{Y \subseteq X, Y \neq \theta} m(Y) \text{ and } \text{Pl}(X) = \sum_{Y \cap X \neq \theta} m(Y)$$

Unlike more traditional probabilistic frameworks that assign mass to singleton hypotheses, DST can assign belief to a *set* of hypotheses. This is useful in the case where the evidence is imprecise and/or incomplete (as is often the case in complex and dynamic network environments). Given two BBAs m_{s_1} and m_{s_2} obtained from evidence sources $s1$ and $s2$ (which may have varying credibility, reliability and completeness), they can be combined via Dempster's rule of combination. For any $X \subseteq \Theta$,

$$m_{s_1 \oplus s_2}(X) = \frac{1}{1-k} \sum_{A \cap B = X} m_{s_1}(A) m_{s_2}(B),$$

where the conflict coefficient k is

$$k = \sum_{A \cap B = \theta} m_{s_1}(A) m_{s_2}(B)$$

However, this combination is only well-defined when sources are not completely contradictory (i.e. $k \neq 1$). The factor $\frac{1}{1-k}$ is a normalising step that distributes the mass associated with conflicting evidence amongst non-conflicting propositions, implicitly removing conflict from the overall result.

5.2.5 Subjective Logic

Subjective Logic (SL) is a probabilistic reasoning framework that extends upon the Dempster-Shafer Theory that explicitly encapsulates belief, disbelief, uncertainty and an optional base rate as an *opinion quadruple* [27, 26]. This opinion is of the form $w_x = (b_x, d_x, u_x, a_x)$ where:

- b_x: The probability mass that x is true
- d_x: The probability mass that x is false
- u_x: Residual probability mass representing incomplete evidence
- a_x: The probability that x is true independent of supporting evidence

The concept of representing incomplete evidence (ignorance) and fusing multiple independent sources of evidence is inherited from the Dempster-Shafer Theory presented in Section 5.2.4, and the interpretation of an opinion in the Bayesian perspective is achieved by mapping opinions into probability distributions. The foundations of SL are expanded in Section 5.4.

5.3 A Comparison of Decision Logic Mechanisms

The core challenge within trust evaluation revolves around the concept of uncertainty, where any resulting decision must be based on evidence that can weaken confidence in an evaluation. Compounding this is at the very nature of an opinion itself; perceived truth is in relation to the evaluating agent, and not necessarily representative of a general, objective truth. It is therefore vital that, in order to model observed phenomena as accurately as possible, a formalism to express uncertainty must be adopted, as well as the ability to assign ownership to individual opinions. Furthermore, the decision logic must explicitly support multi-source evidence and opinion aggregation to cater for the multi-agent nature of the Compute Continuum (CC).

As stated previously, probabilistic logic extends the binary truth to a scale of $[0, 1]$, lending itself to scenarios where probabilities can be reliably estimated. However, the challenge in the kinds of scenarios in which CASTOR operates is that evidence is often incomplete and therefore insufficient to facilitate the assignment of confident single-value probabilities. Rather than providing a single value, it would be more helpful to explicitly define how much uncertainty is associated with a given outcome. This limitation points us towards a more advanced decision-making mechanism that can formalise the expression of uncertainty.

On the other hand, fuzzy logic is better suited towards *degrees of truth* (“vagueness” around concepts, such as “high integrity”, “acceptable availability”). However, in CASTOR, evidence is typically in relation to whether or not a given trust proposition is true or false. In other words, evidence generally either supports, or not, a claim (such as attestation reports and logs), rather than falling into a fuzzy logic category. In

other words, whereas probability is focused on the likelihood that an event occurs or not, fuzzy logic is instead focused on conceptualising the degree of truth of a claim. The nature of trust proposition state spaces considered within CASTOR align more with the former, i.e. distinct, mutually exclusive categories (such as being considered to have acceptable integrity or not, with no in-between). Therefore, fuzzy logic is not enough to accommodate the requirements of the CASTOR framework.

Bayesian Probability is a step towards meeting the demands of the dynamic environment in which the CASTOR framework runs in practice. Prior knowledge, as well as the dynamic updating of belief as new evidence arrives, would in principle make it a good candidate for the trust evaluations envisioned in CASTOR. However, it does not natively support the modelling of multiple independent agents evaluating trust in relation to the same proposition. Furthermore, its mechanisms for fusion of trust opinions are not straightforward, particularly in the case of contradicting and partial evidence.

Dempster-Shafer Theory (DST) is well-suited towards reasoning under the presence of uncertainty, as well as for the fusion of multi-source evidence, and assigns belief masses to sets of possibilities. Furthermore, DST accommodates associative, commutative and non-idempotent combination [28], lending itself towards the combination of sources sequentially and in real-time. However, its reasoning capabilities suffer under heavily-conflicting evidence [47], and doesn't directly account for trust-transitivity which is crucial when trust propagates across nodes, links and paths. In addition, it has been shown that the order in which sources are aggregated can affect the result [16].

Decision Logic Mechanism	Handling Uncertainty	Probabilistic Truth Values	Using Past Evidence	Subjective Beliefs	Fusion in Spite of Conflict	Trust Transitivity
Binary Logic	✗	✗	✗	✗	✗	✗
Probabilistic Logic	✓	✓	✗	✗	✗	✗
Fuzzy Logic	✓	✗	✗	✗	✗	✗
Bayesian Probability	✓	✓	✓	✗	✗	✗
Dempster-Shafer Theory	✓	✓	✓	✓	✗	✗
Subjective Logic	✓	✓	✓	✓	✓	✓

Table 5.1: A visual comparison of the different decision logic mechanisms considered for this project.

Table 5.1 provides a visual comparison of the decision logic mechanisms presented in the sections above. These mechanisms are compared based on core requirements for the CASTOR framework. Firstly, it is apparent that all mechanisms excluding binary logic are capable of dealing with uncertainty in various capacities and through various means. Furthermore, only binary logic and fuzzy logic are incapable of expressing probabilistic truth values. In binary logic, propositions are simply evaluated as true or false as a single value, whereas fuzzy logic exhibits many-valued logic.

Only Bayesian probability, DST and SL are capable of incorporating past evidence. Bayesian logic implements this through the use of prior probabilities. In DST, the BBA encodes everything accumulated so far and can be updated to accommodate new evidence. In SL, new evidence is also considered, for example, through cumulative fusion. Only DST and SL support subjective beliefs from multiple agents concerning the same proposition; however, only SL is well-suited towards fusing these beliefs in the case of highly-contradictory evidence. Also, as explained previously, DST does not model trust transitivity, an explicit requirement of the CASTOR framework – it is often the case that, due to the complex trust networks commonly encountered in practice, trust is evaluated over indirect trust relationships (for example

based on referrals from neighbouring nodes). Therefore, we conclude that SL is the most appropriate decision-logic mechanism for trust assessment in the presence of uncertainty for the CASTOR project. In the following section, we explore the concepts of SL in more detail.

5.4 Foundations of Subjective Logic

As stated previously, CASTOR takes steps towards treating trust as a continuous, context-aware value that is formed upon collected evidence. Evidence can be incomplete or obtained from potentially untrusted sources and as such, there is inherent uncertainty in the overall assessment of trust. CASTOR implements Subjective Logic (SL) as its foundation for reasoning over this evidence, as well as for calculating and analysing Actual Trust Levels (ATLs) and Required Trust Levels (RTLs). SL is well-suited towards decentralised systems, such as those in which CASTOR operates, where evidence is obtained both locally and indirectly, in varying degrees of quality, scope and credibility.

SL is a probabilistic logic framework that expresses trust as an *opinion*, extending classical probability by assigning a given proposition a level of belief (b), disbelief (d) and uncertainty (u), along with an optional base rate (a) that represents belief independently of observed evidence [26]. In addition, SL allows for reasoning over opinions obtained through multiple sources through *fusion*, effectively allowing the aggregation of multiple opinions into a single representation based on the chosen *fusion operator*. Furthermore, SL also supports *discounting*, enabling received opinions to be modulated with respect to the credibility of the reporting entity, using a chosen *discounting operator*.

5.4.1 Subjective Logic Opinions

The foundation of SL is the concept of an *opinion* that encapsulates the amount of uncertainty around the truth of a given trust proposition. As stated above, an opinion is made up of belief masses and uncertainty mass and an optional base rate. A given TAF within CASTOR aggregates a diverse set of trust sources, including secure boot metrics, attestation reports and neighbouring assessments. Rather than trivially merging these opinions (for example by simply performing an average), CASTOR implements SL to formalise the opinion fusion process in a way that takes into consideration the level of *uncertainty* around a claim.

A trust proposition (for example: “Secure Boot for Router A stands”, or “The link between Router A and Router B has integrity”) is encapsulated as a trust opinion of the form $\omega = (b, d, u, a)$, and in an example scenario, one such opinion may be assigned values such as $\omega = (b = 0.3, d = 0.5, d = 0.2, a = 0.5)$. This particular example can be interpreted as being biased towards disbelief with minimal uncertainty. The crucial aspect of an opinion is that the core components, i.e. belief, disbelief and uncertainty, are explicit and measurable.

The notation ω_X^A is used to represent an opinion in SL. Here, X refers to the target variable (i.e. trust proposition) for which the opinion applies, and A refers to the entity that holds that particular opinion. This allows a TAF to maintain a clear and explicit trust opinion about a particular proposition within the trust model. For example, ω_{P1}^{R1} denotes that Router 1 has the opinion ω in relation to trust proposition P1.

Described in more detail in Section 7.2.2, CASTOR maps evidence to the components of a SL opinion using a quantification function. In general, an explicit trust proposition is mapped to belief, disbelief and uncertainty probability masses using the available (and appropriately weighted) evidence.

5.4.2 Binomial and Multinomial Opinions

Formally, an opinion expresses belief about a variable X from a given state space (domain) that represents all the possible states of X. The state space values are exclusive, meaning that only one state value is possible at any time, and exhaustive, meaning that all possible state values are included in the state space. A state space can be binary (with exactly two values) or n-ary, where $n > 2$. A binary state space can be denoted as $\{x, \bar{x}\}$, where $\bar{x}$ is the negation of x. A multinomial state space, on the other hand, can be denoted as $\{x_1, x_2, ..., x_n\}$ where $n > 2$.

In the context of CASTOR, a binary domain represents a trust proposition that can be evaluated as true or false. For example, a binomial opinion formed in relation to the trust proposition "The link between Router A and Router B has sufficient bandwidth" would evaluate to either true (i.e. the bandwidth is sufficient) or false, therefore having a binomial state space. However, treating this instead as a multinomial opinion would allow for more granular evaluation and a multinomial state space, instead evaluating to "High", "Medium" or "Low" to represent how much bandwidth is available.

A binary opinion is used in the case where there are two mutually exclusive outcomes (True or False), and takes the form of the standard opinion $\omega_x = (b_x, d_x, u_x, a_x)$ for the proposition x, under the following constraints:

$$b_x + d_x + u_x = 1$$
$$b_x, d_x, u_x, a_x \in [0, 1]$$

Assuming that a proposition x states that a router has secure boot enabled, the opinion $\omega_{\text{secureboot}} = (0.7, 0.1, 0.2, 0.5)$ can be interpreted as:

- 70% belief that secure boot is enabled,
- 10% belief that secure boot is not enabled,
- 20% uncertainty,
- 50% base rate (i.e. expected probability in the absence of evidence).

This results in an expected probability of $b_x + a_x u_x$, i.e. $0.7 + 0.5 \times 0.2 = 0.8$. This clearly maps to the binary output of True/False, where secure boot is enabled (True) with 80% likelihood, and disabled (False) with 20% likelihood.

Binomial opinions can be represented geometrically in a Subjective Logic triangle, shown in Figure 5.2, with vertices corresponding to belief, disbelief and uncertainty. Any opinion $\omega = (b, d, u, a)$ can be represented as a point within this triangle, constrained by $b + d + u = 1$. The point within the triangle encodes the relative probability masses assigned to each component of the opinion, and intuitively, closer proximity to a vertex implies bias towards that component. Similarly, a central point within the triangle would represent an opinion that is evenly balanced with evidence and ignorance.

One benefit of this representation is that uncertainty, belief and disbelief are visualised clearly, compared to a classical probability value where all aspects are collapsed into a singular value. Another useful note is that Required Trust Level (RTL) values can be represented within the same triangle as an area, which can make it immediately obvious whether or not a specific ATL falls within acceptable criteria. An RTL can be expressed as a constraint on either belief, disbelief, uncertainty, or any combination of the three, that when plotted form a boundary on the triangle. Therefore, the more constraints that an RTL imposes, the stricter the bound within an RTL area becomes. This is expanded upon and visualised in Chapter 8.

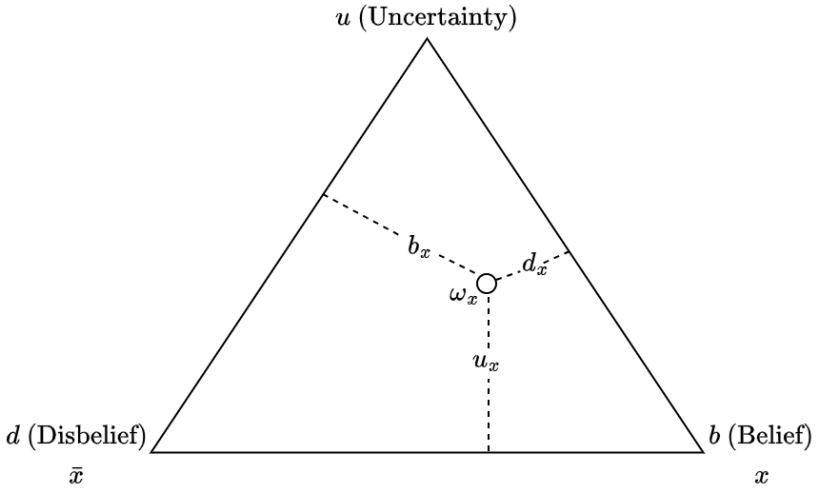

Figure 5.2: The subjective logic triangle for a binomial opinion. An opinion ω_x is shown within the triangle, with its opinion components mapped to their respective belief, disbelief and uncertainty values.

A multinomial opinion is used when there are more than two possible outcomes, such as when the goal is classification or state estimation. A multinomial opinion can instead be represented as $\omega = (\{b_{x1}, ..., b_{xn}\}, u, \{a_{x1}, ..., a_{xn}\})$ for the domain $\{x_1, x_2, ..., x_n\}$, with the constraints:

$$\sum_{i=1}^{k} b_{xi} + u = 1$$

$$\sum_{i=1}^{k} a_{xi} = 1$$

As a brief example, assume the domain $\{\text{Low}, \text{Medium}, \text{High}\}$ is used for estimating available link bandwidth. Then, the opinion $\omega = (\{0.3, 0.4, 0.2\}, 0.1, \{0.5, 0.25, 0.25\})$ can be interpreted as:

- 30% belief that the link has high bandwidth,
- 40% belief that the link has medium bandwidth,
- 20% belief that the link has low bandwidth,
- 10% uncertainty,
- Base rate favours high bandwidth.

In this case, expected probability E is calculated as $E(x_i) = b_{xi} + a_{xi}u$. Therefore, the link is calculated to have low bandwidth with probability $0.3 + 0.5 \times 0.1 = 0.35$, medium bandwidth with probability $0.4 + 0.25 \times 0.1 = 0.425$, and high bandwidth with probability $0.2 + 0.25 \times 0.1 = 0.225$. In this case, we have moved beyond the simple binary output state space, in favour of a more expressive n-ary state space (where in this particular example, $n = 3$).

The geometric representation presented above can also apply to multinomial opinions, with probability masses distributed over more than two mutually exclusive outcomes. However, in this case an opinion encompasses a higher-dimensional space. For a three-state (trinomial) opinion, it is sufficient to represent the opinion within a tetrahedron as in Figure 5.3. However, as the state space grows, the opinion must be represented within an ever-higher dimensional simplex, at which point the clean visual representation begins to break down.

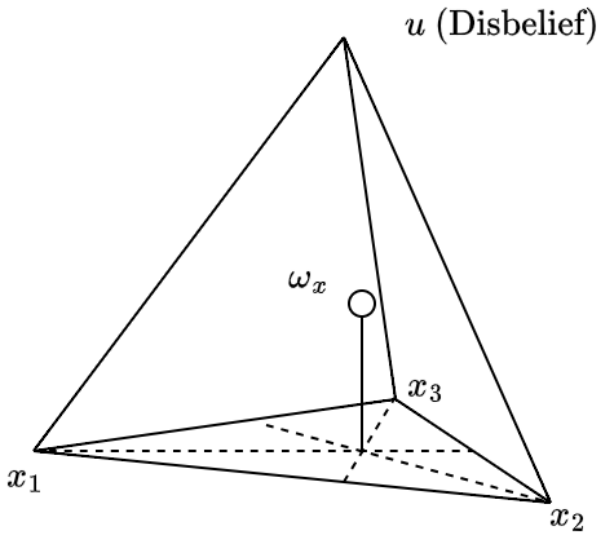

Figure 5.3: A subjective logic tetrahedron for a trinomial opinion. An opinion ω_x is shown within the figure, its position reflecting the balance of belief, disbelief and uncertainty across three possible states. The base vertices represent 100% belief in any of the three states (x_1, x_2 and x_3), with the opposite side representing 100% disbelief in that respective state. Uncertainty is measured based on distance from the base.

A key factor to take into consideration is that the outcome of these decisions (as well as the decision regarding the most appropriate type of state space for a given trust proposition) directly impacts the accuracy of the evaluation of trust. Evidence-based trust quantification directly relies on the availability of supporting evidence, the quality and quantity of which significantly impacts the modelling of uncertainty. Therefore, appropriate state space definition and opinion modelling is crucial to ensure accurate trust decisions are made, directly influencing CASTOR traffic optimisation and, subsequently, which traffic engineering policies are enforced in the network.

5.4.3 Subjective Logic Discounting: Handling Shared Evidence and Opinions

In the case where a TAF receives reported trust assessments and evidence from its neighbours, it must apply a discounting operator that adjusts the received information by an amount that represents how much the receiving agent trusts the reporting agent. This is particularly prevalent in the federated trust assessment modality, where the Global TAF frequently receives trust opinions and evidence from Local TAF agents. This process helps to ensure that information received from less reliable sources is weakened appropriately (usually by increasing the uncertainty component of the SL opinion), allowing referrals from more credible sources to maintain a stronger influence over the decision-making process (implicitly rewarding consistently “good” behaviour).

In other words, discounting facilitates reliable computation of transitive trust in a complex, distributed environment. Formally, this process is captured through *trust discounting*, a fundamental component of SL. When an entity A receives an opinion ω_B^X regarding proposition X, and also holds an opinion ω_A^B on the trustworthiness of B, a discounted opinion ω_A^X can be computed using a trust discounting operator, such that:

$$\omega_A^X = \omega_A^B \otimes \omega_B^X$$

where $\otimes$ represents a chosen discounting operator. Examples of how discounting is used within the CASTOR framework are highlighted in Section 6.1.3, 6.1.4, 6.1.5, 6.1.6, 6.1.7 and 6.1.8.

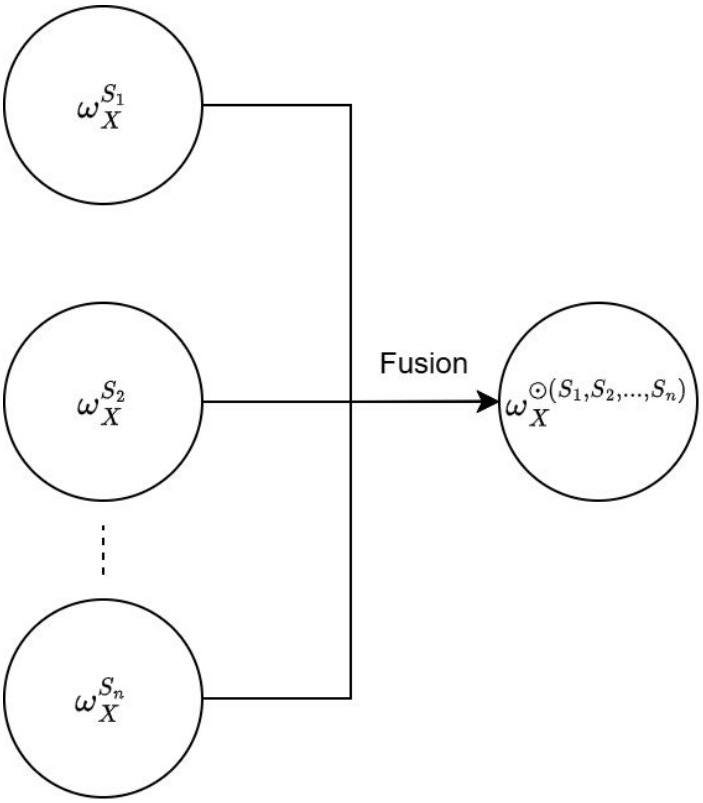

Figure 5.4: Opinions from multiple evidence sources (S) being fused into a single representative opinion.

5.4.4 Subjective Logic Fusion: Handling Evidence and Opinions from Multiple Sources

The fusion of multiple trust opinions is one of the most powerful aspects of SL that enables the formal "merging" of independently-derived trust opinions into a single view, taking into account the credibility of the participating entities. Given two opinions, ω_X^A and ω_X^B, their fusion is represented by the opinion $\omega_X^{A \odot B}$, with $\odot$ representing the chosen fusion operator. This new opinion effectively encapsulates the trust that the conceptually combined agent, $[A, B]$, would have in the trust proposition X [10]. In practice, any n opinions can be fused.

This is most useful from the perspective of the Global TAF; if multiple Local TAF agents are reporting on the trustworthiness of an onboarding node, an appropriate fusion operator allows the Global TAF to derive a single opinion that better represents the overall truth of a proposition with all sources of evidence (even those independent and indirectly related to the Global TAF itself) taken into consideration. However, fusion is not limited to the Global TAF. A Local TAF may fuse trust evidence collected form diverse local sources in relation to the same integrity-related trust proposition, such as secure boot attestations and behavioural runtime analysis. The general process of fusion is illustrated in Figure 5.4. Here, multiple sources of evidence $S_1, S_2, ..., S_n$ form an opinion on the same proposition and a fusion operator $\odot$ is used to derive a single opinion.

Crucially, fusion requires the selection of an appropriate *fusion operator*. Fusion operators should not be confused with traditional logical operators, such as conjunction and disjunction, as they are unique to SL. Examples include *Consensus Fusion, Averaging Fusion, Cumulative Fusion, Weighted Fusion* and *Epistemic Fusion*. The choice of appropriate fusion operator is context-dependent, based on the nature of the evidence itself (and how agreement and disagreement should be handled) and the trust property being assessed. Each reflects different assumptions regarding the relationship between sources of evidence and the quality of evidence itself. Below, we detail some examples of fusion operators.

Consensus Fusion. Consensus fusion aggregates belief and uncertainty by reinforcing agreement

amongst sources and redistributing conflict into increased levels of uncertainty. In other words, strong disparity amongst evidence increases uncertainty rather than artificially averaging out disagreement or incorrectly inflating belief. This methodology has the benefit of clearly modelling conflicting evidence, manifesting as increased uncertainty in the presence of disagreement, making it well-suited towards noisy environments. However, this strength can also contribute to a weakness; dynamic and volatile environments can become oversaturated with uncertainty, weakening the overall decision-making capabilities of the system. Consensus fusion should generally be used when disagreement is expected; it ensures that conflict amongst evidence is appropriately considered and uncertainty is increased.

Averaging Fusion. Averaging fusion calculates a simple mean (average) opinion when all sources are considered with equal weighting. The belief, disbelief, uncertainties and base rates are simply averaged across each opinion, resulting in a much simpler and more "human-interpretable" fusion process. This approach maintains stability in noisy environments, but its effectiveness is limited as the quality of evidence is reduced. For example, a particularly extreme opinion (perhaps calculated in error) can have heavy influence on the resulting averaged opinion. In addition, conflict is effectively "masked", rather than used to modulate trust in the resulting opinion. Furthermore, heavy disagreement amongst sources can lead to an opinion that incorrectly conveys moderate confidence. Averaging fusion is often used when sources of evidence are fairly consistent and when a simple model is preferred.

Cumulative Fusion. Cumulative fusion adjusts belief, disbelief and uncertainty probability mass as new evidence accumulates over time, assuming that new opinions received represent new rather than complementary evidence. This approach more accurately models evidence accumulation over time and can lead to a reduction in uncertainty as more evidence accumulates. However, this approach is more suited towards longitudinal analysis (i.e. measurements over time), making it less useful in cases where an immediate opinion is required in the moment, without requiring future evidence. Cumulative fusion should be used when subsequent opinions represent fresh, independent evidence that is collected over time (such as through the analysis of telemetry and logs) with minimal duplication.

Weighted Fusion. Weighted fusion places emphasis on accounting for variability in source quality and trustworthiness. Explicit weights are assigned to opinions (for example based on metadata or historical observations). This allows for the effective incorporation of reputation based on past behaviour, and lessens the impact of known low-quality and/or unreliable sources. However, weight estimation is non-trivial and must be performed carefully to ensure accurate opinion formation and the avoidance of biased results. Weighted fusion is suited towards scenarios in which sources heavily differ in their reliability and trustworthiness in a way that is quantifiable.

Epistemic Fusion. Epistemic fusion places priority on the handling of uncertainty, opting to preserve ignorance instead of artificially updating belief when available evidence is lacking. This helps to reduce false confidence and lends itself towards more safety-critical decision-making, acknowledging its own lack of knowledge rather than making a best effort in spite of it. However, this conservative approach can make it more difficult to produce opinions with high confidence. Epistemic fusion should be used in the case of sparse and incomplete evidence, when maintaining a view on uncertainty is preferred to forcing a trust decision, for example, in the situation where a false positive (i.e. false confidence) would result in a highly negatively-impacting outcome.

Chapter 6

Trust Modelling in Traffic Engineering Policy Provisioning

6.1 Trust Relationships

In this section, we present the various types of interaction that can occur in a given network topology regarding the formation of trust opinions in relation to the Local TAF(s) and Global TAF. Each scenario incrementally builds upon the previous, starting with the simple scenario of a Local TAF forming an opinion over an atomic trust proposition based on evidence internal to that specific router and culminating in the establishment of a trust opinion based on an entire path, composed of composite trust propositions.

6.1.1 Local TAF Trust Assessment of an Integrity-Related Atomic Trust Proposition

In the base scenario, the Local TAF of Router 1 is forming an opinion on specific aspects of the integrity of the specific router. The breakdown of these aspects is intrinsically linked to the types of evidence available for collection in the target environment. In general, the path profile catalogue offers certain requirements, such as high integrity, and (whilst out of scope for this specific scenario), the Global TAF must collect and form trust opinions relating to the chosen requirements from the Local TAF and fuse them using an appropriate fusion operator to derive the final requirement. The network operator configures the TAFs with respect to the chosen trust policy such that the relevant evidence is collected and the appropriate opinions are formed in relation to the chosen requirements.

For this example, illustrated in Figure 6.1, the Local TAF is configured to collect evidence related to secure boot. In practice, this atomic trust proposition states that “Secure boot in Router 1 stands”, evaluating to True or False. At this stage in the CASTOR project, all trust propositions equate to a Boolean value, however, the possibility of trust propositions evaluating to multiple outcomes (for example CPU utilisation mapping to “high”, “medium” and “low”) remains open for future work. This would extend the ability of the trust assessment framework by allowing more complex reasoning at the cost of increased overhead and complexity from the perspective of subjective logic.

In general, the Local TAF of Router 1 calculates its own Actual Trust Level (ATL) with respect to its integrity, based on direct evidence from internal trust sources, such as secure boot attestation (P1) and is represented by opinion $\omega_{P1}^{LTAF\text{-}1}$. This opinion is subsequently pushed to the Global TAF via a Trust Network Device Interface Security Protocol (TNDI-SP) channel, a process that is discussed in more detail Scenario 2, outlined in Section 6.1.2.

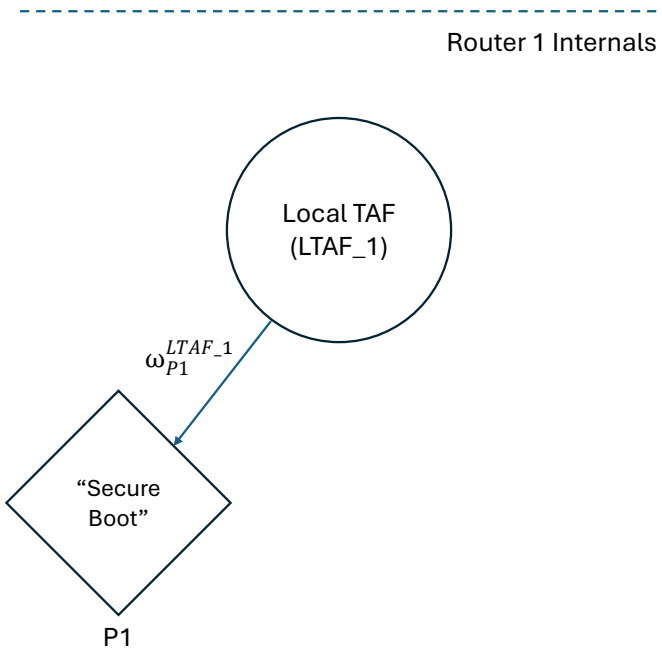

Figure 6.1: Scenario 1 – The Local TAF performs a trust assessment over an atomic trust proposition relating to secure boot.

6.1.2 Global TAF Discounting a Local TAF Opinion

Building on the previous scenario, the ATL derived by the Local TAF is pushed to the Global TAF via a dedicated TNDI-SP channel that bridges the Infrastructure Layer and the Orchestration Layer, as shown in Figure 6.2. Crucially, however, the ATL is not accepted as-is. The Global TAF instead discounts the ATL based on its own trust opinion ($\omega^{G}_{LTAF\_1}$) of Router 1's Local TAF. This process, called "fusion", is a component of subjective logic, and an appropriate fusion operator must be used depending on the given context (an operation that allows the combination of independent trust opinions around the same proposition in the presence of uncertainty)[1] to form a single, accurate and consolidated trust opinion from the perspective of the global TAF agent.

This opinion encapsulates the capabilities and trustworthiness of the local TAF, with respect to providing such evaluations from the perspective of the global TAF. This opinion itself is informed by additional evidence provided by the Local TAF, such as proof of secure launch and correct configuration. This step is crucial in ensuring that the final opinion that the Global TAF derives with respect to Router 1's integrity is adjusted based on the understood credibility of the reporting Local TAF.

In practice, the combination of Scenarios 1 and 2 depicts the overall, albeit simplified, flow of a Global TAF performing a trust assessment in relation to an integrity-related trust proposition, the chosen requirement offered by the path profile catalogue, based on evidence quantified at the Local TAF level. In general, other evidence can be used if a different requirement is specified (such as availability). Indeed, it is often the case that a combination of multiple trust propositions is required to satisfy a given requirement, necessitating the use of composite trust propositions as opposed to those that are atomic. Scenario 3 expands upon the trust assessment of composite trust propositions.

[1] The choice of fusion operator depends on various factors such as the level of overlap and the amount of evidence as well as the desired outcome given the context (such as prioritising agreement or disagreement).

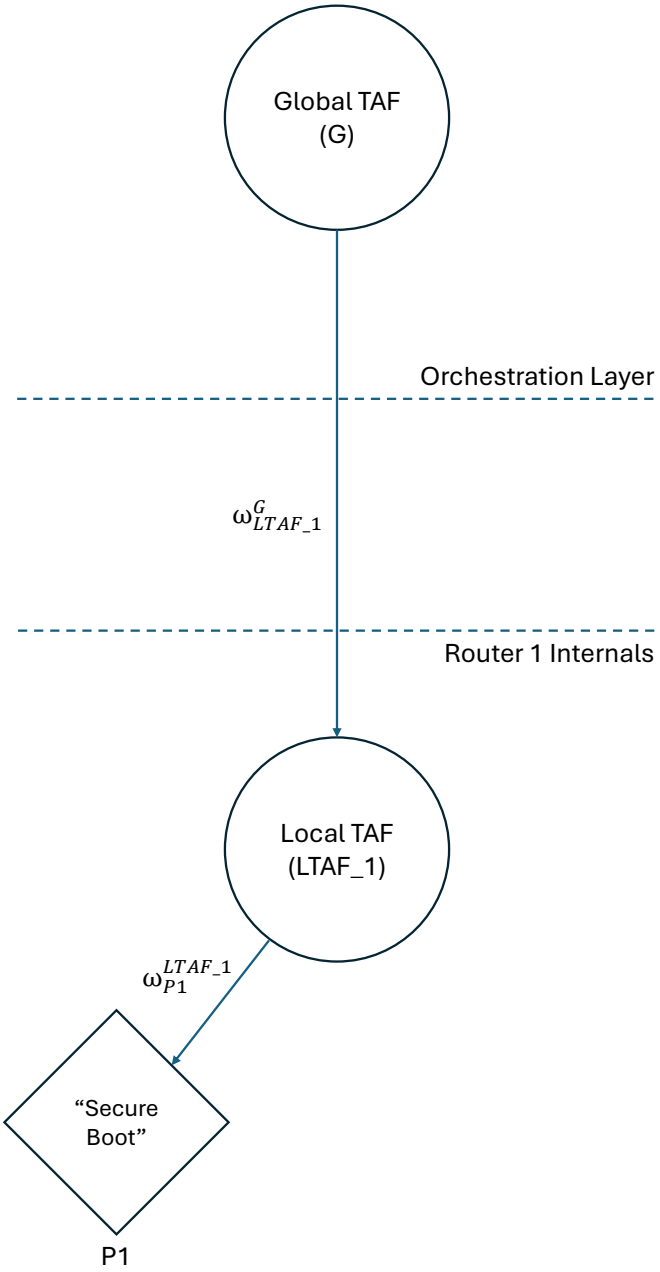

Figure 6.2: Scenario 2 – The Global TAF discounts an opinion received from the Local TAF.

6.1.3 From Atomic to Composite Propositions

In the previous scenarios, the trust propositions formed and evaluated by the Local and Global TAF were strictly atomic. In other words, they could not be broken down into simpler sub-propositions, and they directly map one-to-one to a single source of evidence. However, it is more accurate that the chosen requirements, such as “High Integrity” will be composed of multiple sources of evidence and, therefore, multiple trust propositions, illustrated in Figure 6.3. For example, P3 may state that Router 1 has high integrity, which is formed by the composite proposition: “Secure boot holds on Router 1 AND runtime integrity checks have passed”. Here, it is the responsibility of the Global TAF to receive and “aggregate” independent atomic trust propositions into a single, consolidated trust proposition using logical expressions such as AND and/or OR.

In this scenario, the requirement of “High Integrity” is understood to require evidence of both secure boot attestation and runtime integrity checks. Therefore, we introduce a new atomic trust proposition, P2, stating that “Router 1 runtime integrity checks have passed”. The evidence required to evaluate both P1 and P2 is collected by the Local TAF of Router 1, which forms ATLs for each independently. As before,

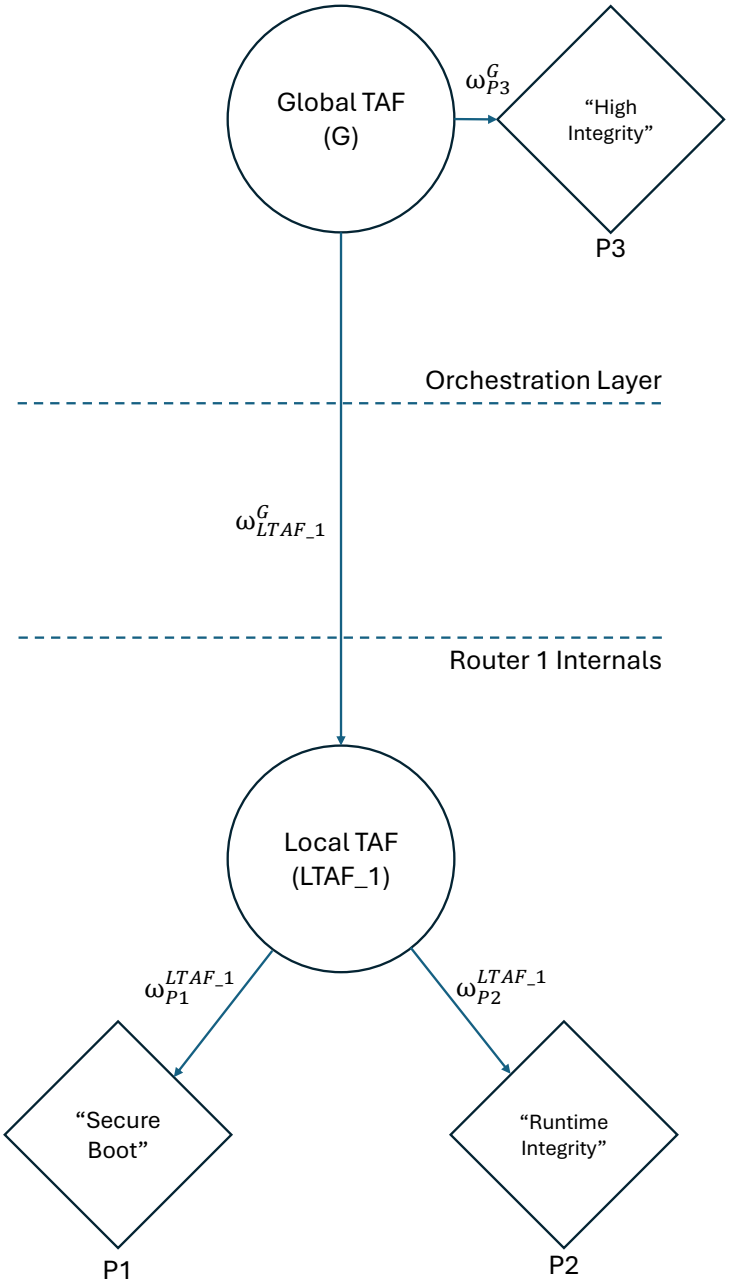

Figure 6.3: Scenario 3 – The Global TAF aggregates a composite trust proposition based on two integrity-related trust propositions from the Local TAF, ultimately forming a composite trust proposition in relation to node integrity.

these propositions are then pushed to the Global TAF, which must again discount them based on its own opinion on the capabilities of LTAF_1 to make such assessments. However, it is now the responsibility of the Global TAF to aggregate each independent opinion into a consolidated view that more accurately conveys whether Router 1 can be considered to have "High Integrity", resulting in a new proposition, P3.

It is also important to note that the opinion $\omega^{G}_{LTAF\_1}$ quantifies the trust of the Global TAF in the ability of the Local TAF to make assessments in relation to integrity. However, if another trust property (such as availability) was to be assessed in parallel, another trust relationship may be required representing the trust of the Global TAF in the Local TAF to make assessments in relation to availability.

Not all evidence can be measured directly by the Local TAF. In such cases, the Global TAF has the possibility to access trustworthiness evidence directly from Trust Network Device Extensions (TNDEs) by leveraging dedicated TNDI-SP channels, allowing the Global TAF to quantify trust opinions directly . This process is explored in Scenario 4.

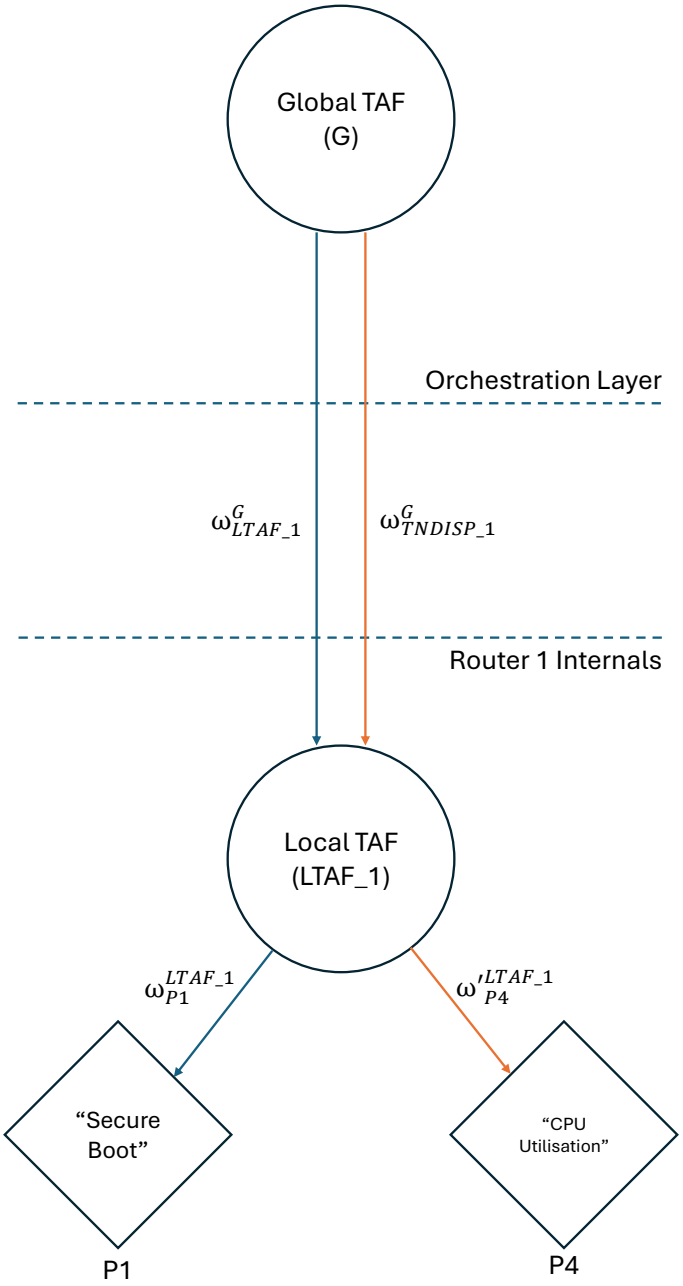

Figure 6.4: Scenario 4 – The Global TAF directly quantifies evidence received via the TNDI_SP channel.

6.1.4 Global TAF Opinion Formation on Evidence from the Orchestrator

In this scenario, the Global TAF is forming an opinion on device-level properties that may not be fully evaluated by the Local TAF, or for which the Global TAF requires direct and/or supplementary evidence. In this case, the Global TAF can make use of established dedicated data channels via the TNDI-SP to securely and efficiently collect trustworthiness evidence directly from TNDEs within a given router. For reference, Figure 6.4 highlights this trust relationship in orange and also contains the Global TAF discounting process of an atomic proposition from Scenario 2 (outlined in Section 6.1.2) so that both workflows can be visually contrasted.

The crucial differentiating factor between this scenario with the previous scenarios is that here, evidence quantification occurs at the Global TAF level rather than at the local level. In other words, evidence is sent directly to the Global TAF via the TNDI-SP channel. However, the Global TAF must take into account the telemetry collection capabilities of this channel, leading to further discounting based on the trust of the Global TAF in the TNDI-SP channel itself. The Global TAF's opinion on the TNDI-SP is derived from

based on an Orchestrator component checking the state of the TNDE, granting use if the corresponding policy is met.

In this scenario, we introduce P4 stating that “CPU utilisation levels do not exceed 50%”. This evidence can be used to form an atomic trust proposition in relation to the Router 1’s availability (e.g. a guarantee that the router does not exceed the chosen threshold of CPU utilisation). The orchestration layer is responsible for configuring TNDEs to collect the relevant traces given the chosen requirements , through the provision of Trust Policies which specify not only the trust models for the Local TAF agents, but also aspects around the evidence-monitoring mechanisms (e.g. the tracing layer and trust sources). The Global TAF can then discount this raw evidence based on its opinion of the TNDI-SP ($\omega^{G}_{TNDISP\_1}$) and subsequently form its own trust opinion on the proposition itself.

Note that no opinion is formed or stored within the Local TAF in relation to P4. This is to ensure that the router software stack is not overloaded with excessive trust calculations. It also enables the orchestrator layer to request evidence directly rather than waiting on the Local TAF agent to evaluate a router-level proposition. This does, however, introduce a trade-off between distributing trust evaluations in a federated TAF modality and resource consumption. Hence, in CASTOR, we provide this capability in the Trust Policy specification. As an example, the approach that we will follow as part of the first version is to configure the Local TAF agents to form local opinions on integrity-related propositions and all other evidence is sent directly to the Global TAF for evaluation.

Raw evidence passes directly through the Local TAF and is sent straight to the Global TAF via the dedicated TNDI-SP channel. However, the trustworthiness evidence constitutes direct observations that are made by the Local TAF agent and its Trust Sources. Hence, from a modelling perspective, the corresponding trust opinion from this evidence should be modelled on top of a direct trust relationship from the Local TAF to the proposition, regardless of whether the actual opinion quantification is taking place at the Local TAF agent or at the Global TAF.

So far, all trust evaluation discussed has been in relation to node-level properties. However, CASTOR allows for the dynamic trust evaluation of entire paths, which are composed of nodes and links. In the next scenario, we explore how the trust assessment framework forms opinions on link-level trust properties.

6.1.5 Global TAF Opinion Formation on Link-Level Trust

This scenario conveys the Global TAF forming a trust opinion of a specific link between two entities in the topology shown in Figure 6.5 (in this case, Router 1 and Router 2). As the Global TAF has full visibility of the underlying network topology and infrastructure, it is possible to collect trustworthiness evidence and form trust opinions specifically for link-level properties of trust.

As a simple example, one situation may involve the Global TAF forming an opinion on the integrity of a given link. In order to do so, the Global TAF must assess the integrity of both nodes present within the link (Router 1 and Router 2). In addition, the property being assessed can also rely on evidence not directly related to the routers themselves. For example, if link availability was to be assessed, the orchestrator can collect telemetry data regarding the link itself, such as available bandwidth. Whilst this evidence is not directly associated with the nodes, it is directly related to the link itself and will factor into the Global TAF’s opinion on the final link trust proposition. The Global TAF must discount this evidence appropriately as in Scenario 4, detailed in Section 6.1.4.

In CASTOR, the modelling of link trust evaluations is modelled without affecting the algebra of agents. Links are not modelled as separate trust objects as this would imply that they are independent agents (in the multi-agent system capturing the infrastructure layer), capable of forming direct or transitive trust relationships. Instead, a trust proposition on a link is treated as a composite trust proposition which can be — eventually — decomposed into atomic trust propositions via logical expressions. Thus, the Global TAF must derive a composite trust proposition concerning a link by combining atomic trust propositions

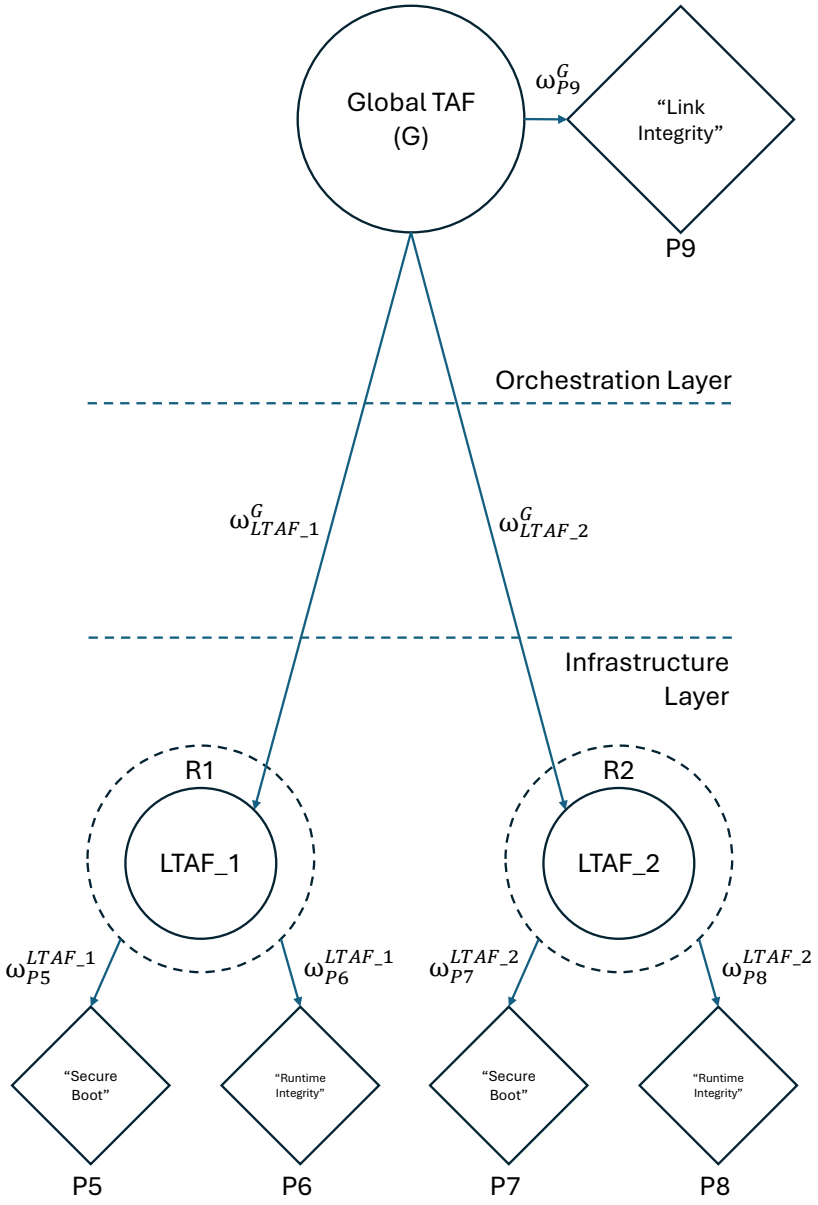

Figure 6.5: Scenario 5 – The Global TAF forms an opinion in relation to link-level integrity.

from independent sources (which as discussed, can include those from the routers themselves based on applicable internal evidence, or those based on the orchestrator's direct observations of network-related metrics).

In this example scenario, the Global TAF individually discounts opinions from both Local TAF agents in relation to device integrity before fusing them into node-level integrity trust propositions. These steps have been omitted from the figure for readability, but can be found in Scenario 3, discussed in Section 6.1.3. Finally, the node-level integrity trust opinions can be fused to form P9. This proposition states that "The link between Router 1 and Router 2 has high integrity", and encapsulates an evaluation of the overall link's integrity.

In the next scenario, we take this workflow one step further by introducing the concept of paths, and explaining how the trust evaluation framework allows the Global TAF to form an opinion on an entire path comprising of several nodes and links.

6.1.6 Global TAF Opinion Formation on Path-Level Trust

This scenario builds upon the previous by introducing the concept of path-level trust. A path is comprised of several nodes and links that ultimately map a route from source to destination. The same logic as in the previous scenarios is applied here albeit at a larger scale. For the sake of readability, we show one of the simplest possible paths in Figure 6.6, involving three routers (Router 1, Router 2, and Router 3). A valid path would be the route from Router 1 to Router 3, which would necessitate traversing two links:

1. Router 1 $\rightarrow$ Router 2
2. Router 2 $\rightarrow$ Router 3

Node-level integrity evidence is evaluated internally to each router resulting in atomic trust propositions relating to the specific trustworthiness evidence that is collected, and the Global TAF discounts each opinion received before fusing them into opinions representing the integrity of each node. Similarly to Scenario 5 (see Section 6.1.5), link-level integrity can now be evaluated by fusing the appropriate node-level integrity propositions. These steps have been omitted from the figure for readability. However, a final additional step is required to subsequently aggregate the link-level integrity assessments into a single path-level integrity assessment, P10, which states “The path between Router 1 and Router 3 has high integrity”.

Importantly, P10 itself may be decomposed into the individual link and node-level propositions that were evaluated during the overall workflow. This ensures granularity when specifying network requirements, such as requiring a path that specifically has secure boot enabled in all of its routers, as well as high levels of availability in all of its links.

6.1.7 Trust Evaluations and Trusted Path Routing (Simple Case)

This scenario introduces the concept of referral trust, i.e. a router forming an opinion of a neighbouring router and forwarding that opinion to the Global TAF. Referral trust is at the heart of the CASTOR trust assessment framework, allowing the evaluation of trust across dynamic network topologies where a direct link between any two participating entities cannot always be guaranteed. As seen in Figure 6.7, a new router, Router N, is onboarding into the topology, and no direct trust relationship currently exists between it and the Global TAF (i.e. the Global TAF currently does not have an opinion related to the new router’s trustworthiness). Thus the Global TAF is, at least initially, unable to make a direct assessment of the new router’s trustworthiness. However, Router 1, a neighbour adjacent to Router N, has already successfully onboarded into the topology and therefore the Global TAF has an opinion on its trustworthiness.

The Local TAF of Router N, LTAF_N, begins to gather attestation reports and other related evidence with respect to the chosen trust property, in this case device integrity, and formulated into the trust proposition P11 which states that “Secure Boot for Router N stands”. This evidence is subsequently shared with Router 1 in the form of a Stamped Passport, allowing Router 1 to form an opinion on the trustworthiness of Router N. Finally, this opinion can be shared with the Global TAF, where subsequent discounting steps take place to modulate the overall ATL relating to Router N with respect to the perceived credibility that the Global TAF has of Router 1. This process allows the Global TAF to indirectly form an ATL regarding the trustworthiness of Router N, implicitly establishing a trust relationship between the Global TAF and Router N.

6.1.8 Trust Evaluations and Trusted Path Routing (Generic Case)

In this scenario, we extend upon the previous one by introducing the concept of fusion. Fusion is a critical component of SL, allowing the Global TAF to form an accurate opinion of a trust proposition in the case

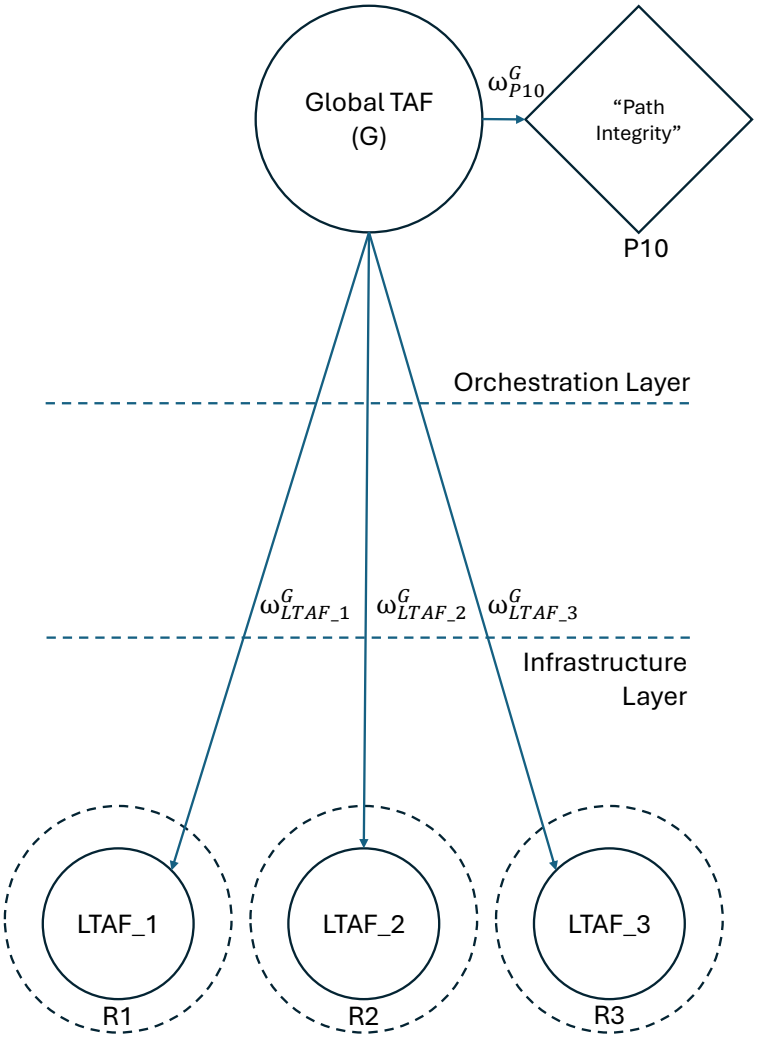

Figure 6.6: Scenario 6 – The Global TAF forms an opinion in relation to path-level integrity.

where it receives multiple opinions (that may be complementary or contradictory) from different Local TAFs that concern the same proposition (for example in the case where they are sharing their opinions on the trustworthiness of an onboarding router). In Figure 6.8, Router N is onboarding into the topology, and has two adjacent neighbouring routers, Router 1 and Router 2. Similarly to the previous scenario, Router N sends Stamped Passports to its neighbouring routers based on P12 stating that "Secure Boot for Router N stands", allowing the already onboarded routers to form opinions regarding the trustworthiness of Router N independently.

However, as Router 1 and Router 2 send their newly formed opinions to the Global TAF, we find ourselves in the situation where the Global TAF receives multiple opinions that concern the same trust proposition. It may be the case that Router 1 considers the new router to be highly trustworthy, but Router 2 (for reasons such as the incorporation of a different trust model, or link-level issues that manifest as a degradation in the quality of evidence) does not consider the new router to be trustworthy. Fortunately, the use of SL allows us to handle this case properly, taking into account the credibility of both Router 1 and Router 2 in order to derive the most accurate ATL possible, all aspects of uncertainty considered. Firstly, the Global TAF must discount each received opinion with respect to the perceived credibility of both reporting routers. Then, the Global TAF fuses these discounted opinions resulting in a single, unified ATL in relation to the trustworthiness of Router N despite there not being a direct link between the Global TAF and the newly onboarded router.

The choice of fusion operator is a highly important consideration to ensure accurate ATL calculation, and is not something that can be decided once and for all scenarios. The optimal choice depends on various

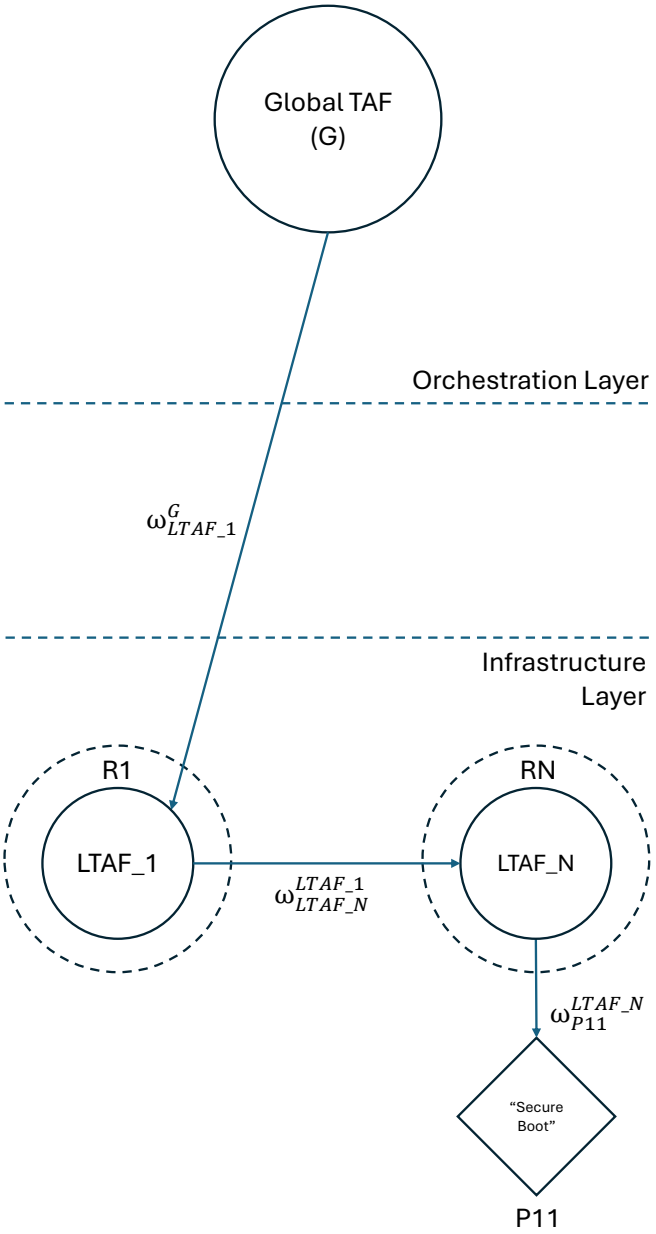

Figure 6.7: Scenario 7 – Router 1 forms an opinion on the onboarding router, Router N, and shares it with the Global TAF where it can be discounted appropriately, based on its opinion of Router 1.

factors such as quality of evidence and the context in which trust is being evaluated. However, generally, the choice of fusion operator determines how agreement and disagreement among sources is managed, and options include cumulative, weighted, consensual and epistemic fusion. Fusion also plays a role in the evaluation of link and path-level trust, as we need to consider how the Global TAF fuses the opinion of each participating entity in a given link or path.

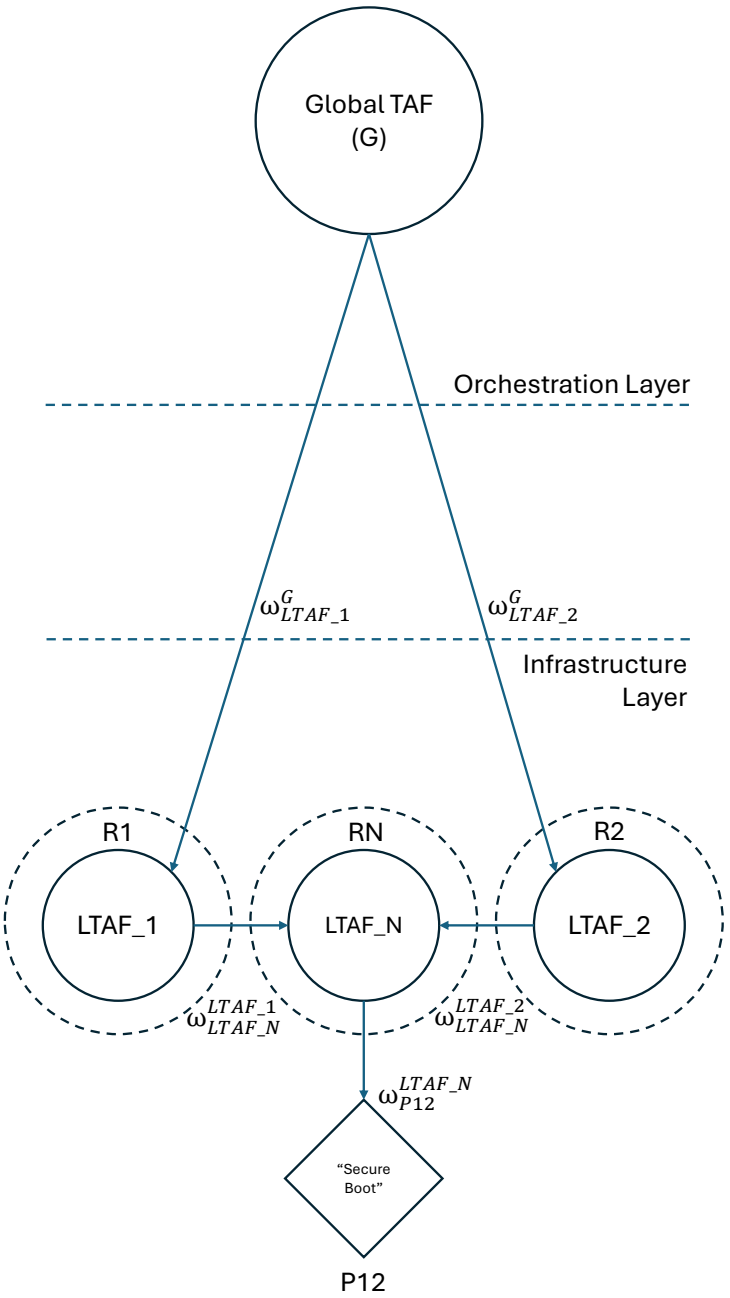

Figure 6.8: Scenario 8 – Router 1 and Router 2 form an opinion on the onboarding router, Router N, and share their opinions with the Global TAF. Here, they can be discounted and fused appropriately, based on the Global TAF's opinions on Router 1 and Router 2.

Chapter 7

CASTOR TAF high-level description

The CASTOR Trust Assessment Framework (TAF) dynamically and continuously evaluates the trustworthiness of key routing-plane components, embedding trust into network traffic engineering decisions. As presented in Chapter 3, CASTOR TAF operates across multiple levels of the compute continuum, enabling trust evaluations from individual network elements up to the orchestration layer. Consequently, CASTOR elevates low-level, element-specific trust assessments into path-level trust characterizations, providing strong assurances to both the Orchestration Layer and the Forwarding Plane (i.e., neighbouring routers). As discussed in Chapter 9, the availability of these service-level trust insights enables the provisioning of trust-aware traffic engineering policies that jointly consider network- and trust-related objectives when enforcing forwarding paths.

At the in-router level, the Local TAF agent is tasked to evaluate the trustworthiness of critical network functions of the router element. Either upon request or periodically, the Local TAF agent leverages the deployed in-router Trust Sources in order to securely collect and report trustworthiness evidence with respect to target trust propositions (e.g., secure boot, runtime configuration integrity, operational integrity) for a specific trust property (e.g., integrity). The secure collection and reporting of this evidence rely on the CASTOR Trusted Computing Base (TCB) which constitutes the backbone of the in-device Trust Network Device Extension (TNDE), as illustrated in D3.1 [7]. Transitioning from the evaluation of static properties (e.g., verifying the secure launch of the TNDE) to runtime properties (e.g., ensuring the integrity of a critical router function's runtime configuration) introduces the need for continuous operation of the CASTOR TAF, in order to capture ongoing fluctuations in router trustworthiness within a given context.

Through the abundance of trustworthiness evidence and local trust evaluations that take place within the router element, CASTOR enables the collection of important low-level trust insights at the Orchestration layer. This culminates in the CASTOR TAF federation which allows the transmission of trust information from the Local TAF agents (and the overall TNDEs) to a central TAF instance, namely the Global TAF. Specifically, the Global TAF is responsible for digesting all available trust-related information coming from the underlying infrastructure layer and deriving the overall trustworthiness of the router elements for a specific trust property. Going beyond that, by employing the appropriate Subjective Logic operators, the Global TAF is able to discount the trust calculations that originate from the in-router TNDE and aggregate them in order to derive trustworthiness claims for network segments, paths, or even entire domains. Overall, the trust evaluations that surpass the in-router boundaries require that the Global TAF has knowledge over the status of the network topology, enabling the incorporation of the relevant trust relationships (and opinions) in the runtime trust calculations.

In what follows, we present the high-level description (see Section 7.1) of a single Trust Assessment instance, shedding light into the internal subcomponents that enable trust evaluations. Subsequently, Section 7.2 discusses the operation of the overall CASTOR Trust Assessment Framework throughout the CASTOR ecosystem, providing initial insights that will guide functional specification of TAF to be documented in D4.2 [8].

7.1 High-level architecture

The Trust Assessment Framework (TAF) is a modular system architecture designed to enable evidence-based evaluation of trust in dynamic, distributed environments. This section introduces the overall architecture of the TAF, therefore, setting the foundation for the subsequent discussion of its two main modalities: the Local TAF agent and the Global TAF.

The CASTOR TAF consists of five tightly integrated functional sub-components that enable trust to be calculated based on runtime and static evidence.

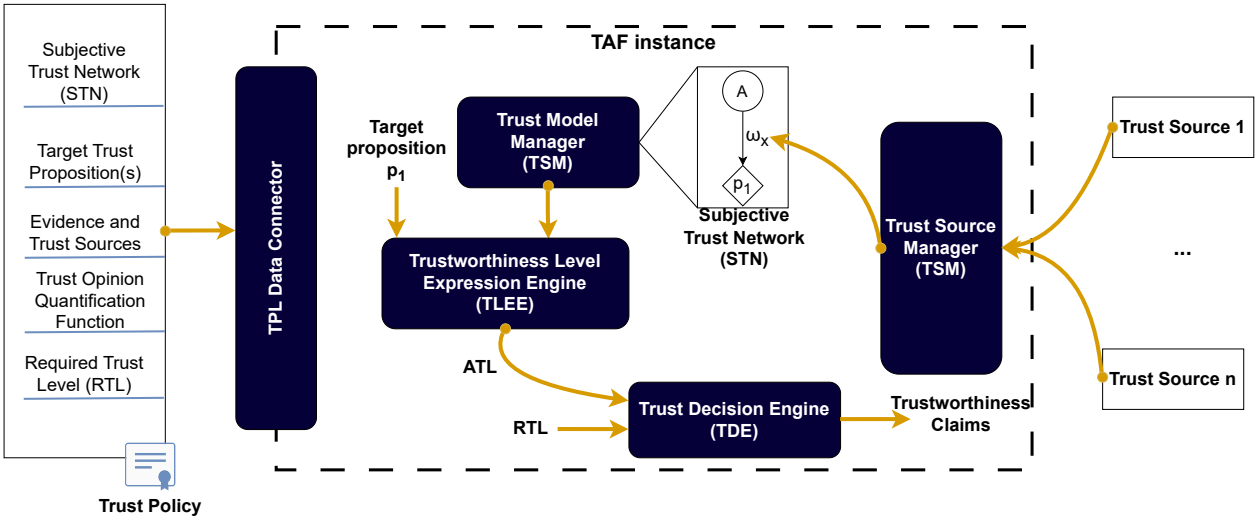

Figure 7.1: High-level architecture of the CASTOR Trust Assessment Framework (TAF)

Trust Model Manager (TMM)

The Trust Model Manager is responsible for instantiating and managing the internal representation of trust relationships. These relationships are modelled as directed graphs of entities and propositions. An entity might be a component of the system. The source of the graph is called ”the agent” and the target is always the proposition to assess. Each proposition represents a logical assertion about some system property—for instance, that a software image is authentic, or that runtime behaviour has not exhibited anomalies.

The TMM supports the composition of atomic and composite propositions. Atomic trust opinions represent the trustworthiness of a single proposition (i.e. variable X). This type of opinion deals with only one specific aspect of trust, and the opinion about this proposition is formed based on direct evidence observed by agent A. Ideally, an atomic proposition is a proposition that cannot be broken down to simpler terms and evidence can either support it or contradict it.

For example, we might combine the following propositions:

- Proposition 1: ”VRouter has started”
- Proposition 2: ”vRouter is operational”
- Proposition 3: ”vRouter has been detected with vulnerability x”
- Proposition 4: ”vRouter forwards messages in less than $1\mu s$”

Composite propositions express higher-level assertions that are evaluated as function of several atomic propositions using logical composition rules.

In practice, the TMM loads these models from policy-driven templates. This allows the system to adapt its reasoning logic based on operational context or threat models. For example, in a high-assurance environment, the model may include detailed runtime behaviour checks and multi-source redundancy, whereas in a lightweight deployment, a simplified trust graph may suffice.

Trust Source Manager (TSM)

The Trust Source Manager serves as the integration layer between raw evidence generators and the trust reasoning engine. It registers and coordinates the trust sources for a given trust model. Trust sources include static measurement tools (e.g., secure boot logs), dynamic monitoring components (e.g., host intrusion detection), remote attestation protocols, or behavioural telemetry systems.

The TSM abstracts the variability of these sources by transforming their outputs into structured, interpretable trust claims. It validates the authenticity and freshness of the data, handles conflict resolution among redundant sources, and converts inputs into normalized trust opinions.

Additionally, the TSM manages the lifecycle of evidence collection: triggering periodic checks, subscribing to real-time feeds, and managing failure fallbacks when sources become unavailable or compromised. It ensures that the trust graph remains populated with current and relevant evidence.

Trustworthiness Level Expression Engine (TLEE)

The Trustworthiness Level Expression Engine is the computational core of the TAF. It takes the normalized trust opinions generated by the TSM and evaluates them within the trust graph defined by the TMM. This involves recursively combining trust opinions across the graph structure to compute the trust value of higher-level propositions.

The engine operates on Subjective Logic algebra, enabling it to manage, fuse, discount, and propagate trust as a first-class property of trust. This is crucial in environments where partial or conflicting evidence is common. For example, if two attestation reports partially disagree, or if one source is delayed, the TLEE does not discard the result but incorporates it with appropriate weighting (i.e., depending on the operator used, it might increase uncertainty or put more weight to the most certain one.).

The TLEE outputs Actual Trust Levels (ATLs) for each target proposition defined in the policy. Each ATL contains the computed trust of the proposition and the source traceability that led to the computed result. This enables transparent audit and explanation of trust decisions.

Trust Decision Engine (TDE)

Once ATLs are computed, the Trust Decision Engine evaluates them against the Required Trust Levels (RTLs). The comparison may end to a binary result (trust granted or denied) by means of a threshold or an ordinal result (high, medium, low).

The TDE supports configurable policies for dealing with borderline or uncertain cases. For example, in safety-critical systems, an ATL below threshold may trigger immediate mitigation, whereas in resilient systems, it might prompt redundancy or escalation.

The TDE outputs actionable results: whether a proposition (which might be about a system or component) is considered trustworthy or not. These results are consumed by the orchestrator layer for the computation of the trusted path.

7.2 TAF within the CASTOR framework

The functionalities of the CASTOR TAF span the entire operational lifecycle of a network domain. On the one hand, it captures aspects of the network element lifecycle, including the secure onboarding of a new element into the topology and its continuous evaluation to ensure its trustworthiness throughout its operational lifespan. On the other hand, the CASTOR TAF is involved in operations related to the fulfilment of new service requests that must satisfy specific trust-related requirements, as well as the continuous evaluation of the relevant network path, thereby ensuring service assurance throughout its operation. All these operational aspects are captured in the overarching phases of the CASTOR Architecture in D2.1 [5]. The relevance of these phases to the overall Trust Assessment Framework is summarized below:

- **Preparedness phase**: The owner of the infrastructure identifies the path profiles will be offered by the underlying infrastructure. This phase includes the network- and trust-related characteristics that each path profile should offer. Trust characteristics can be expressed in the form of requirements that the underlying router elements must adhere to.
- **Proactive phase (Router on-boarding)**: This phase captures the process that a router must follow in order to enrol in the overall topology. This on-boarding process involves the execution of an initial trust assessment task that can evaluate whether the router meets the minimum (integrity) guarantees posed by the network operator of this infrastructure. Part of the on-boarding process is also the establishment of the interactions with the neighbouring routers to ensure the continuous monitoring and assessing of trustworthiness across the topology.
- **Reactive phase**: This phase characterizes the trust assessment operations that need to be running when services are established and workload traffic is forwarded over network paths. With these policies, the control plane (i.e., via the CASTOR Facility Layer) is able to have an up-to-date view of the network and trust properties of all the routers in the underlying network segments.

7.2.1 TAF in Preparedness phase

During the design phase, the network operator needs to identify the path profile catalogue that will be offered to potential service providers that want to use this CASTOR-enabled infrastructure to serve their (critical) application workloads from point A to point B (see D5.1 [6]).

Each path profile aims to offer a set of network- and trust-related guarantees that should satisfy the agreed Service Level Agreement (SLA) and Security SLA established between a service provider and the network operator. These guarantees are expressed in relation to a set of trust properties that the offered path profiles should provide. A preliminary set of trust properties to be considered for routers, links, and/or paths include Integrity, Availability, Confidentiality, and Robustness.

Based on these trust properties, the network operator is able to form trust propositions that express the behaviour of the entities participating in an envisioned path. Trust propositions are built in relation to one or more trust properties and one or more trust objects. For instance:

- *Integrity* of router's software stack
- *Integrity* of the data flowing a particular link between two routers
- *Integrity* of the link between two routers
- *Availability* of a router to server traffic through its interfaces
- *Availability* of the data flowing through a particular link

- *Availability* of a link in terms of bandwidth.

The definition of the trust-related guarantees of a path profile is expressed in the form of constraints defined over one or more trust propositions. As further elaborated in Chapter 8, these constitute the Required Trust Level (RTL) constraints, which may characterize routers, links, and/or the overall path that will be selected once a path profile is deployed.

Depending on the intrinsic characteristics and capabilities of each router and its identified risk level, it is possible that there may be different RTL constraints that different types of routers may need to satisfy in order to serve a common trust guarantee for a trust property. This is intrinsically linked with the risk posture of each type of router to be employed in the underlying infrastructure as well as the (residual) risk that a network operator is willing to accept in order for the entire infrastructure to operate. Consequently, a network operator may require different types and amounts of evidence from routers with varying characteristics.

Based on the above, the specification of the offered path profile catalogue involves the identification of all the trust-related information that will enable the overall Trust Assessment Framework to establish and maintain the trust characterization of the entire infrastructure layer throughout its operational lifecycle. This information includes the following aspects that need to be specified (reflected also in the left part of Figure 7.1):

- Type of router
- Phase in the router lifecycle: On-boarding (installed and registered), Idle (powered on, no traffic), Serving traffic (forwarding packets), Updating (applying patches), Under configuration (changing policies).
- Target Trust propositions to be evaluated
- Decomposition of trust propositions into observable, atomic trust propositions.
- Trust Sources that enable the evidence collection for the assessment of the atomic trust propositions.
- Quantification function for the derivation of each atomic trust proposition.
- Logic that evaluates the measured Actual Trust Level (ATL) of each target trust proposition against the Required Trust Level (RTL) constraints.

The aforementioned pieces of information comprise a ”Trust Policy” which, when enforced in a TAF agent, guides the configuration and execution of all trust calculations within a specific context and scope. This enables continuous monitoring of trust, ensuring that the underlying infrastructure meets the required guarantees throughout the different phases of the CASTOR framework. Based on this breakdown, it becomes clear that there are different Trust Policies per type of router, per phase (e.g., on-boarding of a router, runtime phase where a router servers one or more path profiles).

Once the Trust Policies have been specified - expressing both the minimum requirements for a router to enrol the topology and the guarantees that each path profile shall offer - it is possible to instantiate the Global TAF agent running in the control plane (i.e., in the orchestration layer). This involves the enforcement of all the Trust Policies designed for the Global TAF agent.

In the Global TAF agent, we consider target trust propositions on routers, links, and paths. These will allow us to identify the trust profile that each trust object has during runtime (each entity in the topology is attributed with a network and trust profile; they are used for the optimization process). The set of trust propositions may evolve over time as new routers get enrolled in the topology (or get detached from

the topology). Similarly, the decomposition of target trust propositions may be different over time as the number of routers (and thus associated trust propositions) fluctuates. At the same time, the envisioned target trust propositions may be decomposed to a set of atomic trust propositions of a router and a set of recommendations from its neighbouring routers (i.e., Local TAF agents sharing recommendations to the Global TAF agent). To capture this dynamic behaviour and to accommodate for recommendations between neighbouring routers, all relevant trust opinions are managed via a trust model per policy.

In CASTOR, the selected trust model representation to capture all the dynamic trust relationships and their trust opinions during runtime is a ”Subjective Logic Trust Network”. Initially (i.e., when router is enrolled in the topology), the instantiated trust models are empty. Subsequently, as routers get on-boarded, the corresponding trust relationships are dynamically introduced in the trust model and the respective trust opinions are formed and maintained throughout the router lifecycle.

In the following section, we discuss concrete examples of trust propositions that are formed from the early phases of a routers secure enrolment until its inclusion in the topology. Following a bottom-to-top approach, we start with the description of the trust propositions in the Local TAF agents (i.e., per router). Initially, we describe the on-boarding aspects and then the runtime phase. In the latter stage, we also go through the composite target trust propositions that refer to the Global TAF agent.

7.2.2 TAF in Proactive phase

As a first step, we consider the phase in which a new router is onboarded in the managed infrastructure layer. In CASTOR, routers may be deployed as virtualized functions by the network service orchestrator or operate as conventional hardware devices. In either case, we assume that all in-router CASTOR components—namely, the CASTOR TCB, Local TAF agent, and Trust Network Device Extensions—run as processes alongside the router’s core functionality.

When the Local TAF agent of a router is launched as part of the on-boarding phase (details on the CASTOR secure on-boarding protocol are presented in D3.1 [7]), it connects to the CASTOR DLT to collect a Trust Policy and carry out its initial trust assessment process. Specifically, this refers to the on-boarding Trust Policy that is associated with the specific characteristics of that particular router (e.g., type of hardware/firmware, software stack, applied security controls).

As part of the on-boarding Trust Policy, the Network Service Orchestrator is able to evaluate that the router is able to provide the necessary trustworthiness guarantees that are dictated by the network operator. As part of the enforced Trust Policy, the Local TAF agent receives the decomposition of the target trust proposition into atomic trust propositions. These atomic trust propositions can be measured by the Local TAF agent through the available Trust Sources that are supported by the router. Hence, the resulted trust opinions characterize functional trust relationships between the Local TAF agent and the atomic trust propositions. *In principle, this specification of the atomic trust propositions is intrinsically linked to the available evidence that we can collect and the threat model that we take into consideration*.

The Local TAF agent uses its own Trust Sources and forms opinions about its atomic trust propositions. From a trust modelling standpoint, the Local TAF agent, along with its Trust in-device Trust Sources, form a single analyst entity (i.e., a single agent) that is able to evaluate the trustworthiness of the in-router trust properties. This allows the Local TAF agent to collect various types of evidence and form direct (i.e., functional) trust relationships with the in-router trust propositions that need to be evaluated. Ideally, each trust opinion is mapped to a single type of evidence collected from a single trust source (i.e., we should avoid duplication in the available evidence for a specific trust proposition). The quantification function that maps the available evidence to a trust opinion is provided in the Trust Policy. However, it is worth noting that there may be the a case in which multiple types of Trust Sources provide evidence that refer to the same atomic trust proposition (e.g., the runtime integrity of the software stack of a router may stem from evidence coming from the Attestation Source, and the FSM Source; both presented in D3.1 [7]). In such

cases, the Trust Policy should provide the quantification function to aggregate the evidence and derive the trust opinion for that particular atomic trust proposition.

In order to better depict the trust opinion quantification process, the following example is presented: As part of the on-boarding Trust Policy, the orchestrator may request the Local TAF agent to provide an assessment on the integrity of the router. In this phase, this target trust proposition p_1 is decomposed to two atomic trust propositions: i) router is securely launched, and ii) the operational profile of critical router functionalities has not been tampered. The former atomic trust proposition, $p_{1,1}$ is associated to evidence coming from the Attestation Source and its evidence regarding the configuration integrity of the router. Whereas, the router operational assurance is monitored via the FSM Source which is able to process the relevant operational traces and identify any violation in the transition of the designed device-based finite state machine. This forms the second atomic trust proposition $p_{1,2}$.

Based on the aforementioned example, the trust opinion over an atomic trust proposition ($p_{1,1}$ or $p_{1,2}$) is equivalent to the trust opinion of the single trust relationship between the analyst node (i.e. the Local TAF agent) and the atomic trust proposition. An example of how we could quantify the receipt of positive or negative trustworthiness evidence regarding a router's configuration integrity and map it to its corresponding trust opinion ω_x for $p_{1,1}$ (similarly for $p_{1,2}$) is shown below:

$$b_x = \frac{r_x}{W + r_x + s_x}, d_x = \frac{s_x}{W + r_x + s_x}, u_x = \frac{W}{W + r_x + s_x}, \tag{7.1}$$

where r_x corresponds to the number of evidence supporting the trust proposition, s_x any other evidence against it, and W is a weight to adjust the level of uncertainty even if evidence is available.

Based on this, the TLEE is invoked in order to derive the ATL value for the target trust proposition, aggregating the available trust opinions for the atomic trust propositions. Here, we may have different approaches on how this could designed, depending on the strategy that we want to follow:

- Use Subjective Logic (SL) Logical operators to aggregate the trust opinions for the available atomic trust propositions. E.g., ATL(p_1) = $\omega_{1,1}$ AND$\omega_{1,2}$. However, one concern is that these opinions are "owned" by the same analyst node, so it may not be the most suitable approach.
- No SL operator is applied in the Local TAF agent. All the SL operators are taking place by the TLEE subcomponent of the Global TAF agent. In order to make any local-based trust decision, the respective RTL constraints should be mapped to the trust opinions for the atomic trust propositions. E.g., The belief threshold for $\omega_{1,1}$ is 0.8, while the uncertainty for $\omega_{1,2}$ should not exceed 0.3.
- Use other quantification functions for combining all relevant evidence into a single SL opinion that characterizes the target trust proposition.

Regardless of the strategy to be employed, we may result in one or more atomic trust propositions each of which is characterized by its own ATL value. As already mentioned, these ATLs will be evaluated against the RTL constraints enclosed in the onboarding Trust Policy. This allows the Local TAF agent to derive a trust decision about the trustworthiness of a router (e.g., with respect to its integrity) during onboarding phase. This trust decision is communicated in a secure manner to the CASTOR Global TAF at the orchestration layer which depending on the (local) trust decision outcome evaluates the level of trust that can be placed on this element.

Once enrolled, the Network Service Orchestrator proceeds with the configuration of the necessary communication channels with the newly added router (see D3.1 [7]). It is worth noting here that the Trust Policy to be enforced after the on-boarding phase, may also enable the processing of trustworthiness evidence coming from neighbouring routers (e.g., through Stamped Passports specified in IETF Trusted Path Routing [4]). **Based on the above, we conclude that the main types of evidence that a Local TAF agent may process are the ones shown below:**

- Trustworthiness evidence regarding security properties of the router (e.g., has secure boot, has correct configuration, has properly established application keys MacSec protocol). These claims are derived from a set of traces collected by the CASTOR Tracer layer and processed by the Attestation Source running as part of the CASTOR's Trust Network Device Extensions in the router.
- Trustworthiness evidence pertaining to more complex, runtime guarantees for a particular behaviour of the router target environment (e.g., has updated the routing table correctly, has correctly spawned a new process for managing the iptable). To monitor such claims, it is necessary to extract the traces that are relevant to the target router behaviour, and evaluate the operational assurance of critical router functions and whether any violation has taken place. These traces are processed by the Finite State Machine Source running as part of the CASTOR's Trust Network Device Extensions in the router.
- The aforementioned trustworthiness evidence allows a Local TAF agent to form trust opinions with respect to the state of its own router (ego router). In addition, a Local TAF agent of a router that is enrolled in the topology may also receive trustworthiness evidence coming from a neighbouring router's trust sources (e.g., Stamped Passport of a router presenting an attestation quote as part of an implicit attestation process).

7.2.3 TAF in Reactive phase

From a Local TAF agent's perspective, there is no major difference in the sequence of actions between the proactive and reactive phases; only the Trust Policy may change to provide sufficient trust guarantees (i.e., beyond integrity) for each one of the path profiles.

As mentioned in the introduction, the main purpose of the Global TAF agent is to continuously assess the trustworthiness of the entire infrastructure layer and therefore enable the selection of optimal paths for the offered path profiles that satisfy the requested application workloads. From the perspective of the global Trust Policy specification, this is primarily reflected in the identification of target trust propositions that correspond to the trust guarantees offered by each path profile. As already highlighted, this guarantees may be node-, link- or path- centric. Hence, the Global TAF should be capable of characterizing the trustworthiness of the underlying topology graph for each offered trust property. This attribution of nodes, links and paths in the topology unlocks the optimal selection of paths (see Chapter 9) that need to be established to serve the application workloads with the agreed network (SLA) and trust (SSLA) guarantees.

In general, the Global TAF agent should be able to cope with a multitude of trust propositions referring to various trust properties. First and foremost, the Global TAF receives evidence and quantifies an ATL value for atomic trust propositions of each router element. In a second stage, it may use logical SL operators to derive the target (composite) trust propositions that best characterize the requirements for the offered path profiles. Regarding the possible atomic trust propositions, we consider the following ones:

- *Device-level trust propositions related to integrity as a trust property*. In this case, the Global TAF agent takes into consideration the trust evaluations carried out by each Local TAF agent, discounted with the trust opinion of the Global TAF agent to the Local TAF agent. In order for the latter trust opinion to be formed, the Local TAF agent provides additional evidence pertaining to its ability to provide evaluations for the corresponding trust property, namely integrity. One example of such evidence refers to the fact that the Local TAF agent has securely launched and running in an isolated environment with the expected configuration. In fact, this additional piece of information may be part of the trust report that the Local TAF agent shares with the Global TAF agent. Thanks to this additional evidence, the Global TAF agent is able to model a trust relationship with the corresponding Local TAF agent, and form an opinion about its capability to make evaluations. In principle, the fact

that the Local TAF agent may show evidence about its capability to provide evaluations pertaining to integrity, does not guarantee that it is also capable to make evaluations for other trust properties. This behaviour is in line with the types of additional evidence that we envision as part of the trust report. For instance, we could incorporate additional evidence pertaining to the tracing capabilities of the Trust Network Device Extensions that are deployed together with the Local TAF agent.

- *Device-level trust propositions related to other trust properties*. It is possible that the Local TAF agent does not provide evaluations (i.e., trust opinions) for all the device-level trust propositions that need to be considered as part of the trust guarantees of a path profile (e.g., to avoid overloading the router resources with trust-related tasks). In this case, the Global TAF agent may establish a data channel through the Trust Network Device Interface Security Protocol (TNDISP) to securely collect trustworthiness evidence directly from the TNDEs of a router. For example, one such trust proposition may refer to the availability of a router, quantified in relation to the real-time CPU utilization.

- *Link-level trust propositions*. The Global TAF agent runs at the orchestration layer, providing full visibility over the underlying infrastructure. This means that it is possible to collect trust-related metrics regarding link properties. One simple example is trying to assess the trustworthiness of a link's availability with respect to the available bandwidth that it has in a given moment. As in the previous case, such metrics need to be securely extracted (i.e., through the Tracing Layer) and shared via a dedicated TNDISP-enabled data channel (details on the in-router TNDE architecture are presented in D3.1 [7]).

With these atomic trust propositions, the Global TAF agent is able to maintain a full overview of the trust characterization of the entire network topology. In order to construct this global perception, the Global TAF agent needs to discount the trust opinions shared by the Local TAF agents with the trust opinion that the Global TAF has over them. Of course, depending on the requirements, different SL Discounting operators may be used (i.e., favouring uncertainty vs. disbelief). On top of that, in the case where the Global TAF agent receives recommendations about the trustworthiness of a node (i.e., in terms of integrity), the Global TAF agent should be able to fuse the various trust opinions in order to derive its own ATL for the device-based trust propositions. Similarly, depending on the fused trust opinions, different SL Fusion operators need to be selected - e.g., selection of the appropriate fusion operator depending on the dependencies between the trust propositions and the related evidence.

So far, we have explained how the Trust Model of the Global TAF agent could provide global perception for all the atomic trust propositions that are relevant for each device and link. This perception is maintained and updated dynamically in order to depict an accurate view of the trustworthiness of the infrastructure layer. However, the final step would be to map all the atomic trust propositions into the path profile constraints. For this purpose, we consider that for the specification of the trust requirements of a path profile it may be necessary to assess more complex trust propositions that take into consideration multiple trust properties at a time; e.g., the premium path profile requires high availability and high integrity at a link or even a path level.

In order to realize such statements, one has to identify a decomposition process that will translate a composite trust proposition, say for the premium path profile, to a logical function of atomic trust propositions. This function is realized via the use of SL Logical operators and is provided as part of the Trust Policy enforced to the Global TAF agent. By adopting this approach, the internal Global TAF process would follow the steps below, following the universal TAF architecture of Figure 7.1:

1. Global Trust Model is instantiated and contains an updated view of all the trust opinions that characterize all the specified trust propositions.

2. Invocation of the ATL calculation process. This can be triggered either synchronously (i.e., in a request-response fashion) or asynchronously (i.e., every X seconds according to the Trust Assessment Request).

3. The Trust Level Expression Engine (TLEE) takes as input the overall SL trust network and the target trust propositions along with their decomposition function to the available atomic trust propositions.

4. For each atomic trust proposition, the TLEE isolates the corresponding (sub-) network that is relevant to the specific context. Then, it aggregates all the trust opinions that appear in the trust relationships of the network; either through discounting or fusion of the relevant trust opinions.

5. Eventually, the TLEE applies the decomposition function to aggregate the trust opinions for the relevant atomic trust propositions into a single ATL value for each target trust proposition. These ATL values indicate the trust opinion of the Global TAF agent that a trust object (e.g., a device, a link, a path) has the guarantees (i.e., trust profile) required to serve a specific path profile.

6. The final step is to attribute each trust object with a specific trust profile. To do this—i.e., to label each entity (router, link, or path) in the infrastructure layer—we must verify that the requirements defined in each path profile are met. This verification occurs in the Trust Decision Engine, which compares the resulting ATL values for the target trust propositions against the RTL constraints defined in the Trust Policy. This comparison enables the classification of each entity into a particular trust profile. On one hand, this process allows nodes, links, and paths to be labelled with the trust characteristics of the topology. Combined with the network profiles constructed by the orchestrator, the optimization engine can recommend sets of optimal paths that satisfy each path profile. On the other hand, once a path is identified and enforced, the Trust Decision Engine can flag cases where any participating entity has been downgraded to an unacceptable trust profile. Eventually, as detailed in D5.1 [6], this allows the Orchestration Layer to apply mitigation actions and switchover to other available network paths that meet the path profile requirements.

7.3 Trust Engineering in the Case Study

Inspired from the Case Study network topology of Section 4.3.1, in this section we want to provide some initial findings on how we envision to approach the trust modelling process. In what follows, we explore a trust model snippet that corresponds to the trust relationships that a Global TAF should assess in order to characterize the trustworthiness of different propositions within a network segment comprising of a set of interconnected network elements.

Based on this setup, and following the Trust Relationships that have been identified in Chapter 10, Figure 7.2 illustrates a Trust Model Snippet in the form of a Subjective Logic Trust Network that captures the trust relationships of the Global TAF to a pair of neighbouring router elements. Through this example, we are able to flesh out the key challenges that will guide the functional specification of the Trust Assessment Framework, to be documented in D4.2 [8].

We consider two trust properties: integrity and availability.

The Trust Policy enforced in the enrolled Local TAF agents enables the evaluation of two trust propositions pertaining to integrity:

$p_1(i)$: The assessed router i has secure boot, and $p_2(i)$: The assessed router i has configuration integrity during runtime.

The Local TAF agent - associated with each router enrolled in the topology - evaluates the two atomic trust propositions for each own target environment and shares its ATL values to the Global TAF agent via the Telemetry API. As part of this trust report, the Local TAF agent bundles evidence about itself being

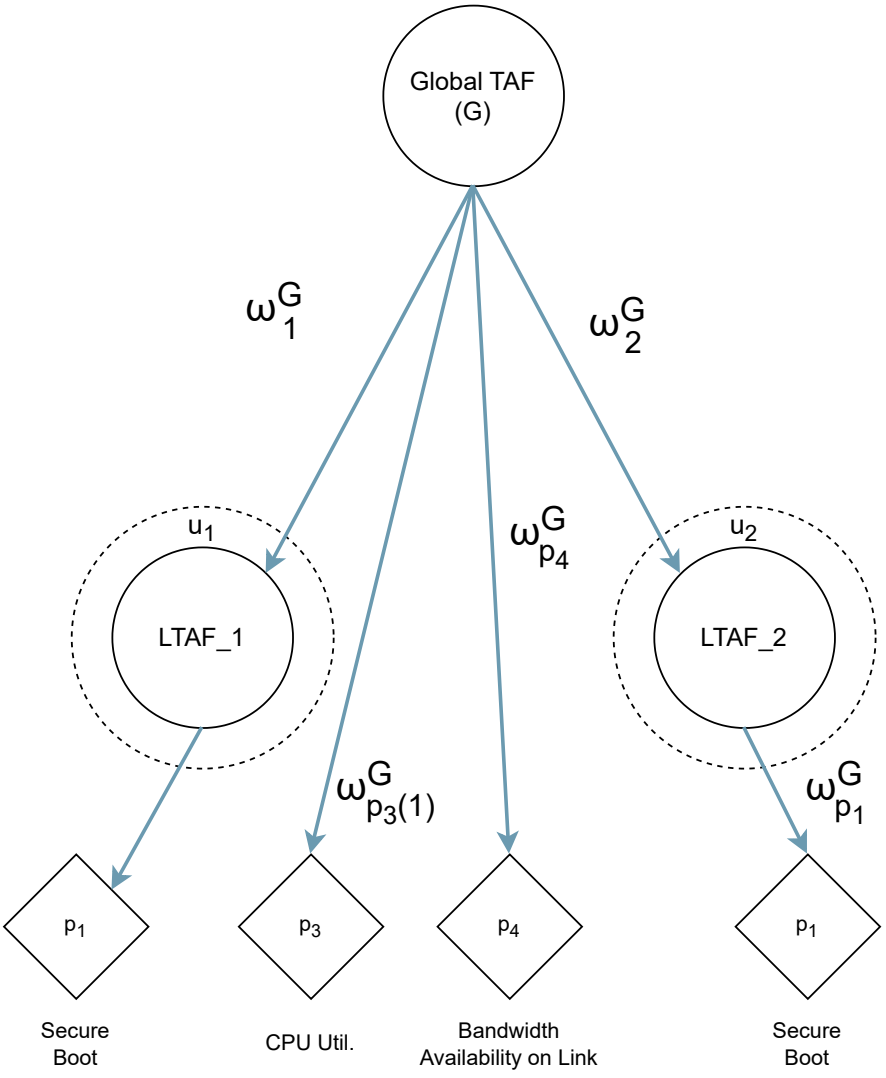

Figure 7.2: Exemplary Trust Model instance on the Global TAF agent; 2 interconnected routers.

at a correct state. This includes also the correctness of the tracer functionalities that contributed to the formation of these local opinions.

In parallel to these evaluations, the orchestrator configures the TNDEs of each router and collects CPU utilization data through a TNDISP-enabled data channel. This allows the global TAF agent to form the following trust proposition:

$p_3(i)$: The assessed router i does not exceed 60 per cent of each CPU utilization.

Similarly, using the same technique, the Global TAF agent is able to securely acquire evidence pertaining to the availability of the established links between the routers. This allows for the definition of the following atomic trust proposition:

$p_4(x, y)$: The established link between routers x and y has always available bandwidth of 1Gbps

With these propositions in mind, we present an instance of the global trust model considering two routers and one link between them. From this SL Trust Network we model the following trust opinions:

1. Trust opinions from Local TAF agents to the respective atomic trust propositions. One example is $\omega^1_{p_1(1)}$ which captures the trust opinion of Router 1 on the proposition $p_1(1)$.
2. Trust opinions from Global TAF agents to the respective atomic trust propositions. One example is $\omega^G_{p_3(1)}$ which captures the trust opinion of the Global TAF agent on Router 1's proposition $p_3(1)$.
3. Trust opinions from one Local TAF agent to one its neighbouring routers. Depending on the attestation policy, the Stamped Passports may provide evidence with respect to an atomic trust proposition or the capabilities of a neighbouring router to form opinions with respect to trust property. Hence, we consider to different trust relationships in this recommendation scheme. First, example is $\omega^1_{p_1(2)}$ which captures the trust opinion of the Router 1's Local TAF agent on Router 2's secure boot integrity- i.e., $p_1(1)$. Secondly, when the Stamped Passports provide evidence about the correct setup of the neighbouring router's Local TAF agent and its TNDEs, the example ω^1_2 captures the trust opinion of Router 1's Local TAF agent on Router 2's capability to assess it's own target environment.

This showcases how the high-level requirements captured in the trust profile associated with a path profile are translated to enforceable Trust Policies in the routers. In fact, the Trust Policies are also adjustable according to the risk-aware derivation of the RTL constraints; from the example we observe that Router 2 which is identified to have more vulnerabilities that compromise its integrity needs to provide more evidence and attain higher ATL than Router 1.

Based on the aforementioned snippets we can also infer the target trust propositions that the Global TAF agent needs to calculate in order to evaluate whether the example topology (i.e., router 1 is connected to router 2), satisfies the path profile trust requirements defined in the trust profile.

By inspecting the trust model above, we consider the following composite trust propositions:

c_1: The assessed router 2 has integrity with respect to its secure boot and its runtime configuration

c_2: The link between routers 1 and 2 has integrity as long as both routers have integrity

c_3: The link between routers 1 and 2 has availability as long as both routers do not exceed 60% of their capacity and the link has 1 Gbps bandwidth for further consumption

c_4: The link between routers 1 and 2 can support trust profile "very-high"

These composite trust propositions can be decomposed to the available trust propositions, The equations for their corresponding ATL trust opinions are shown below (without the included recommendations; $\otimes$ corresponds to a discounting SL operator, $\oplus$ corresponds to a fusion operator.):

$ATL(c_1) = \omega^G_{c_1} = (\omega^G_2 \otimes \omega^2_{p_1(2)}) \oplus (\omega^G_2 \otimes \omega^2_{p_1(2)})$

$ATL(c_2) = \omega_{c_2}^G = (\omega_1^G \otimes \omega_{p_1(1)}^1) \wedge (\omega_{c_1}^G)$

$ATL(c_3) = \omega_{c_3}^G$ = similar to $ATL(c_2)$ but using the trust propositions regarding availability

$ATL(c_4) = \omega_{c_4}^G = \omega_{c_2}^G \wedge \omega_{c_3}^G \wedge \omega_{c_4}^G$

Eventually, the ATL decompositions are provided as input parameters to the TLEE component of the Global TAF agent. In fact, they could be encapsulated within a single decomposition function that constructs the target trust proposition c_4, which is relevant for evaluating whether the topology satisfies the path profile requirements. Following this approach, the output would be a single ATL value, namely $ATL(c_4)$. The total number of final ATL values produced by the Global TAF agent depends on the Path Profile requirements, i.e., the Trust Policy specification.

Through the definition of these ATL equations as part of an overall Trust Policy, we are able to identify three core challenges when determining the functional characteristics of the CASTOR Trust Assessment Framework.

Challenge 1 From the aforementioned equations that need to be processed at the Global TAF level, a key question that arises relates to the availability of the information on the internal trust opinions. Specifically, by inspecting the example of $ATL(c_1)$, it becomes clear that the Global TAF needs to have access to the opinion on the Local TAF capabilities in router 2, but also on Local TAF agent's perception on the trust proposition $p_1(2)$. *As part of the federation of TAF agents (see Chapter 3), the latter trust opinion can be derived either by having the Local TAF agent sharing its computed trust opinion, or by having the TNDE-collected evidence been sent directly to the Global TAF (and perform the trust opinion quantification at the global level). The realization of this form of federation poses significant challenges (e.g., synchronization of trust models between agents, compatibility of semantics and trust operations used between agents) that will be further explored in D4.2 [8].*

Challenge 2 In addition to the previous challenge, this ATL formulation helps clarify another critical consideration, namely the trustworthiness of Local TAF agents. As explained in D3.1 [7], the Local TAF agent operates in isolation (i.e., within an enclave environment) from the rest of the in-router software stack. As part of the first version of the CASTOR TAF, the Local TAF agent is therefore considered a fully trusted application. This assumption directly impacts the transitivity of trust from the perspective of the Global TAF, as the ω_i^G opinions are assigned a full belief value, namely $\omega_i^G = (1, 0, 0)$ for binomial opinions. *However, relaxing this trust assumption within the in-router TNDE environment introduces the additional challenge of providing verifiable evidence of the correctness of the Local TAF agent as part of its trust reports before sharing them with any external entity (e.g., the Global TAF).*

Challenge 3 Finally, throughout this chapter, we consider the federation modality in which Local TAF agents provide evaluations to the Global TAF regarding the trustworthiness of their associated in-router behaviour. However, through the exchange of trustworthiness evidence across the forwarding plane as part of the IETF Trusted Path Routing paradigm, it is possible for a Local TAF agent to receive trustworthiness evidence about a neighbouring router (see the trust relationship scenario in Section 6.1.8). While this mechanism enables Local TAF agents to share opinions about their network vicinity (e.g., the integrity of adjacent network elements), it also introduces potential pitfalls that may bias the final opinion derived by the Global TAF, particularly due to dependency and circularity in trust propagation, whereby recommendations about a given router are influenced—directly or indirectly—by that same router or by mutually dependent neighbours. The aforementioned bias does not necessarily imply that a rogue vRouter is able to provide falsified trust evaluations; under an honest-but-curious model, it is also possible that a vRouter (or multiple colluding vRouters) selectively withholds or forwards trust information to the Global TAF, thereby influencing its perception. This concept of circular dependencies arise when trust evaluations are mutually dependent. For example, the Global TAF assesses two vRouters by requiring the opinions of $LocalTAF_1$ and $LocalTAF_2$, while each Local TAF requires the other's evaluation to complete its own. This mutual dependency prevents the trust values from being independently established (e.g., Transitive trust relationships $GlobalTAF \rightarrow LocalTAF_1 \rightarrow LocalTAF_2$ and $GlobalTAF \rightarrow LocalTAF_2 \rightarrow LocalTAF_1$). In adversarial settings, this effect can be further exacerbated by Sybil-like attacks, in which a compro-

mised node creates and controls multiple trust profiles [43], allowing it to present strategically crafted or inconsistent evidence to different neighbours. *As a result, even if this issue may appear to be addressed by Challenge 2, bias can still be introduced into final trust evaluations, causing the Global TAF to aggregate opinions that are not truly independent and leading to overestimated belief, reduced uncertainty, or skewed ATL evaluations. CASTOR therefore explicitly targets this challenge, aiming to ensure the by-design construction of trust models that are resilient to circular dependencies and unintended reinforcement effects among different TAF agents.*

Through this case study, it becomes clear that the Global TAF constitutes a foundational element that is able to collect observable evidence from the underlying infrastructure element, therefore realizing the overarching federation modality that is envisioned in CASTOR. In this context, the Global TAF is able to collect trustworthiness evidence and Local TAF agent opinions in order to form composite trust propositions that reflect the trust requirements expressed by the network operator and offered to any service provider. The detailed functional specification of the CASTOR TAF, and its first release as part of the overall CASTOR framework is documented in D4.2 [8].

Chapter 8

Risk-aware RTL derivation

8.1 The need for RTL values

In zero-trust network architectures, the fundamental principle is that no entity, whether internal or external, should be implicitly trusted. This principle requires continuous verification and assessment of all network participants before granting access to resources or relying on their provided data. However, this raises a critical question: what level of trustworthiness is sufficient for an entity to be considered reliable in a specific operational context?

The CASTOR architecture addresses this challenge through a dual-metric approach:

- Required Trustworthiness Level (RTL): A design-time specification that defines the minimum trustworthiness threshold required for an entity or data item to be considered acceptable for use.
- Actual Trustworthiness Level (ATL): A runtime assessment that quantifies the observed trustworthiness based on collected evidence and behavioural analysis.

The trust decision is fundamentally based on comparing these two metrics such that an entity or data item is deemed trustworthy if and only if its ATL meets or exceeds the established RTL for the relevant context.

RTL values serve multiple critical functions within the CASTOR framework. They provide concrete, quantifiable criteria for trust decisions during path computation and data validation. Moreover, RTL values translate organizational risk tolerance and security requirements into operational trust thresholds. Different network functions, data types, or operational scenarios may require different RTL values, reflecting varying criticality and risk exposure. Importantly, unlike ATL which is dynamically assessed at runtime, RTL is determined during system design based on threat analysis, risk assessment, and functional requirements.

The derivation of appropriate RTL values is therefore a critical component of the CASTOR trust assessment framework, as it establishes the baseline against which all runtime trust evaluations are compared.

8.2 Risk Assessment Methodology Overview

The derivation of RTL values in CASTOR is fundamentally grounded in systematic risk assessment. Risk assessment provides the analytical foundation to determine the level of trustworthiness required for different network entities, data flows, and operational contexts. By quantifying the potential impact and likelihood of security threats, risk assessment enables the translation of abstract security requirements into concrete trust thresholds.

8.2.1 Risk Assessment Foundations

Risk assessment in network security typically evaluates threats against critical system properties, including:

- **Integrity**: The assurance that data and system states have not been modified by unauthorized parties.
- **Availability**: The guarantee that network services and resources remain accessible when needed.
- **Authenticity**: The verification that entities and data originate from claimed sources.
- **Confidentiality**: The protection of sensitive information from unauthorized disclosure.

For each property, risk assessment methodologies evaluate:

1. **Asset identification**: Determining which network components, data flows, or functions require protection.
2. **Threat analysis**: Identifying potential attack vectors and threat actors that could compromise the assets.
3. **Vulnerability assessment**: Analyzing weaknesses that could be exploited to realize threats.
4. **Impact evaluation**: Quantifying the consequences if a threat is successfully executed.
5. **Risk determination**: Combining likelihood and impact to produce overall risk levels.

The connection between risk assessment and RTL derivation follows a fundamental principle: *higher risk scenarios demand higher trust requirements*. When an asset or data flow faces significant threats with severe potential impacts, the RTL for entities that interact with that asset must be correspondingly stringent. This ensures that only sufficiently trustworthy entities can participate in high-risk operations, effectively using trust as a risk mitigation mechanism.

CASTOR's approach to RTL derivation is designed to be methodology-agnostic. In the literature and standardization efforts, there are different risk assessment frameworks, including Threat Analysis and Risk Assessment (TARA) for automotive cybersecurity (ISO/SAE 21434) and domain-specific organizational risk frameworks. The RTL derivation mechanism can incorporate risk assessments from various sources, provided that they produce quantifiable risk levels that can be systematically mapped to trust requirements. This flexibility allows CASTOR to adapt to different operational domains and organizational risk management practices while maintaining a consistent trust-based security model.

8.3 RTL Expression Framework

Calculating the Required Trust Level (RTL) presents several fundamental challenges that must be addressed to enable effective trust-based decision making in dynamic network environments. This section outlines the RTL expression framework, identifies key challenges in RTL derivation, and establishes the foundation for the risk-based approaches discussed in subsequent sections.

8.3.1 RTL as Decision Thresholds

Unlike ATL, which represents a concrete trust assessment at a specific point in time, RTL functions as a set of decision thresholds rather than a fixed trust opinion. In the context of subjective logic, RTL establishes constraints that determine whether a runtime trust assessment is sufficient for operational requirements. Specifically, RTL defines:

- b_{RTL}: the *minimum required belief* threshold
- d_{RTL}: the *maximum acceptable disbelief* threshold
- u_{RTL}: the *maximum acceptable uncertainty* threshold

These thresholds do not form a traditional subjective logic opinion where $b + d + u = 1$. Instead, they represent independent constraints that must all be satisfied for a trust decision to succeed. An ATL opinion at runtime is deemed acceptable if and only if it meets all three conditions simultaneously: $ATL_b \geq b_{RTL}$, $ATL_d \leq d_{RTL}$, and $ATL_u \leq u_{RTL}$.

Figure 8.1 illustrates this threshold-based approach within the subjective logic triangle. The three RTL thresholds (b_{RTL}, d_{RTL}, u_{RTL}) each define a constraint region within the triangle:

- The **belief threshold** b_{RTL} (green solid line) defines the minimum acceptable belief level, creating a region (green triangle) where $b \geq b_{RTL}$.
- The **disbelief threshold** d_{RTL} (blue solid line) defines the maximum tolerable disbelief, creating a region (blue trapezoid) where $d \geq d_{RTL}$.
- The **uncertainty threshold** u_{RTL} (red solid line) defines the maximum acceptable uncertainty, creating a region (red trapezoid) where $u \geq u_{RTL}$.

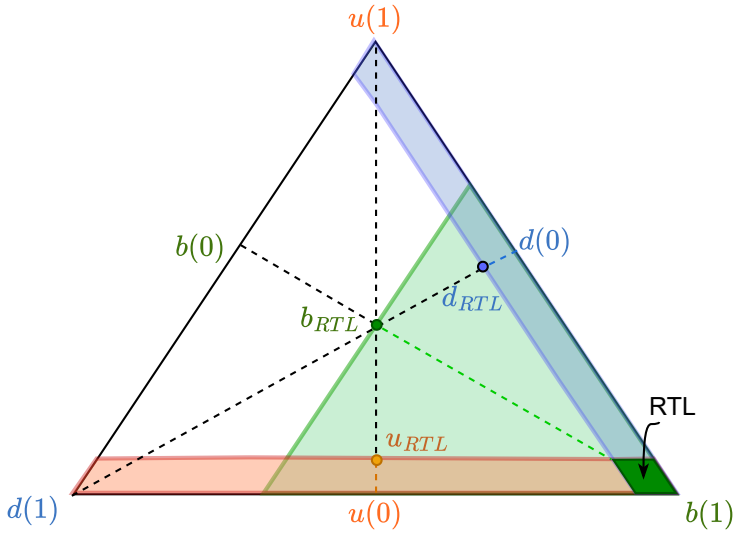

Figure 8.1: Graphical representation of RTL thresholds within the subjective logic triangle. The RTL constraints for belief (b_{RTL}), disbelief (d_{RTL}), and uncertainty (u_{RTL}) define an acceptable region for trust decisions rather than a fixed opinion point.

The intersection of these three constraint regions forms the acceptable trust region (shown as the green diamond-shaped area in the figure). Any ATL opinion that falls within this region simultaneously satisfies all three RTL thresholds and is therefore considered trustworthy for the given operational context. In contrast, ATL opinions outside this region violate at least one threshold constraint and fail to meet the required trust level. This geometric interpretation clearly demonstrates that the RTL defines an *acceptable region* rather than a target point, allowing flexibility in how trust requirements can be satisfied while maintaining rigorous threshold enforcement.

It is important to note that while CASTOR employs subjective logic as the primary framework for trust representation, the RTL concept is not inherently tied to this specific formalism. RTL fundamentally represents a set of constraints that must be satisfied by runtime trust assessments, and these constraints can be expressed using alternative trust models or probabilistic representations depending on the operational context. The critical requirement is that RTL and ATL share compatible semantics, which is that they must quantify trustworthiness using the same underlying dimensions and metrics to enable meaningful comparisons.

The RTL Calculation Problem

RTL values are defined for a given trust evaluation context and scope, expressing the level of requirements that the ATL must satisfied in order for a trustee entity to be considered trusted. These requirements are intrinsically linked to the trustee's risk posture within the defined context and scope of the evaluation. The risk posture itself is established through a thorough and continuous risk assessment process, which systematically identifies assets, the vulnerabilities affecting them, and the threats that an adversary may realize by exploiting those vulnerabilities. This process provides a structured understanding of the exposure faced by the trustee and forms the foundation for deriving meaningful RTL values.

As highlighted in significant risk assessment frameworks (i.e., both in the ENISA's *EU Risk Management Toolbox* [19] and in NIST's Special publication 800-39 on *Managing Information Security Risk: Organization, Mission, and Information System View [34]*), the two main factors that affect the risk posture of an asset capture both the potential consequences of a successful threat realization and the feasibility of such an event occurring, taking into account the adversary's capabilities and the effectiveness of existing controls. Specifically, the primary factors influencing risk are defined as follows:

- **Impact:** The severity of consequences if a threat is successfully posed against an asset or function.
- **Likelihood:** The feasibility or probability that an attacker can successfully execute a threat, considering the complexity of the attack, the required resources, and the existing security controls. As we examine in Engineering Story-II, this may further depend on the number of intermediate steps an adversary must undertake to execute a cascading attack, potentially involving the exploitation of multiple sequential vulnerabilities affecting one or more assets within the network topology.

Through these parameters, it is possible to identify the security controls that must be enforced at each network element (i.e., allowing the mitigation of the critical risks in the topology), as well as the trustworthiness evidence that must be measured throughout the operational lifecycle in order to derive the ATL value for the required target trust propositions, bound to a specific context and scope. At the same time, as explained below, the overarching risk analysis determines the RTL values that must be satisfied by the measured ATL values in order to establish and maintain a trust relationship during the runtime operation of the topology.

Semantic Consistency between RTL and ATL

A primary challenge in RTL derivation is ensuring **semantic consistency** between RTL and ATL. Since trust decisions are made by comparing ATL with RTL, both metrics must be expressed using the same trust model and quantification semantics. Specifically:

- The **belief** component must represent the same property in both RTL and ATL. A mismatch in what belief represents—for instance, integrity versus availability—would render the comparison invalid.
- The **disbelief** quantification must align, ensuring that the negative evidence observed at runtime is consistently mapped to the risk factors that determine the maximum acceptable disbelief levels in RTL.
- The **uncertainty** measurement must be compatible - RTL uncertainty (reflecting incomplete knowledge during design) and ATL uncertainty (reflecting missing evidence at runtime) must quantify the same property: lack of information about the trustworthiness of an entity.

Without semantic alignment, comparing $ATL \geq RTL$ becomes an invalid operation, as metrics would be measuring fundamentally different properties despite using the same mathematical representation.

8.3.2 Evidence Weighting and Trust Opinion Formation

The challenge of semantic consistency extends to the operational level of trust assessment. When runtime evidence is collected and monitored for specific threats, each piece of evidence contributes to the formation of trust opinions. The weights assigned to different evidence types directly influence how belief, disbelief, and uncertainty evolve during ATL computation.

This raises a critical question: *how should evidence weights be determined such that the resulting ATL semantics align with the RTL semantics derived from design-time risk assessment?*

Consider a concrete example: let's suppose that risk assessment identifies a threat of unauthorized software execution on a network node, with high impact (system compromise enabling lateral movement) and high likelihood (known exploits exist for the platform). This assessment should inform two aspects:

1. **RTL threshold**: The high risk translates to a stringent belief requirement (e.g., $b_{RTL} = 0.8$) for trusting the node.
2. **Evidence weighting**: During runtime, evidence from secure boot verification—which directly addresses this threat—should carry substantial weight in ATL calculation. If secure boot verification succeeds, it provides strong positive evidence increasing belief significantly. Conversely, failed secure boot verification should dramatically increase disbelief, as it indicates the high-risk threat is potentially active.

In contrast, a lower-risk threat (such as non-critical configuration drift with minimal impact) should result in both lower RTL requirements and proportionally lower evidence weights. Evidence related to this threat would have less influence on the final ATL value.

The challenge lies in systematically deriving evidence weights from risk assessments such that the proportional influence of different evidence types on ATL mirrors the relative risk priorities established during RTL derivation. Without this alignment, runtime trust decisions may emphasize low-risk factors while underestimating high-risk indicators, undermining the effectiveness of trust-based security mechanisms.

8.3.3 Attack Paths and Cascading Effects

A significant complication arises when considering **attack paths** and **cascading effects**. Threats rarely exist in isolation - successful exploitation of one vulnerability can enable subsequent attacks on other components, creating multi-hop attack chains. The feasibility and impact of such cascading attacks may differ substantially from individual threat assessments.

Current risk assessment methodologies (such as TARA) typically evaluate threats individually or through manually-constructed attack scenarios. However, in dynamic network environments with complex topologies, the attack surface and feasible attack paths evolve continuously. This introduces several unresolved questions:

- How should RTL values account for cumulative risk from cascading attacks rather than only individual threat assessments?
- When attack paths span multiple network hops or domains, how should the RTL for intermediate nodes reflect their role in enabling downstream attacks?
- How can RTL derivation incorporate dynamic attack feasibility that changes based on runtime network state and topology?
- Should nodes- that serve as potential pivoting points in attack paths- have inherently stricter RTL requirements, even if they do not directly handle critical assets?

These challenges become particularly acute in trust-aware routing scenarios, where path selection decisions depend on comparing node trustworthiness (ATL) against requirements (RTL), yet the RTL itself should ideally reflect the cascading risk implications of including that node in a path.

8.3.4 Open Questions for Advanced RTL Derivation

The challenges outlined above motivate several research questions that must be addressed to develop robust RTL derivation methodologies for dynamic trust-aware networks:

1. **Semantic Alignment**: What formal framework can ensure that RTL and ATL quantifications are semantically compatible and enable valid comparison across different trust dimensions?
2. **Evidence-Risk Mapping**: How can risk assessment outputs be systematically mapped to evidence weights such that runtime trust assessment reflects design-time risk priorities?
3. **Attack Path Integration**: How can attack path analysis and cascading risk propagation be incorporated into RTL calculation, particularly in dynamic network topologies where attack paths evolve continuously?
4. **Multi-dimensional RTL Optimization**: Given that different methodologies may yield different RTL values for belief, disbelief, and uncertainty components, what principled approach should determine the final RTL constraints? How should trade-offs between stringent belief requirements and tolerant uncertainty bounds be balanced?
5. **Temporal Dynamics**: How should RTL adapt when network topology changes, new vulnerabilities are discovered, or threat landscapes evolve? Should RTL be recalculated in real-time or at discrete intervals?

6. **Automatic Feasibility Calculation**: How can attack feasibility, currently manually assessed by security administrators in frameworks like TARA, be automatically quantified using techniques such as Markov Chains for temporal attack progression and Monte Carlo simulation for probabilistic what-if scenarios?

Addressing these questions requires moving beyond static, manually-configured risk assessments toward dynamic, topology-aware RTL derivation that can account for cascading attacks and evolving threat scenarios. The detailed mechanisms for such advanced RTL calculation, including the integration of Markov Chain models and Monte Carlo methods for automatic attack feasibility quantification, will be elaborated in Deliverable D4.2. The following section presents existing example approaches to illustrate current state-of-the-art methodologies and identify the specific gaps that CASTOR aims to address.

8.4 Risk-based RTL Calculation: An example approach

While the CASTOR framework is designed to be methodology-agnostic regarding RTL derivation, it is useful to examine existing approaches that demonstrate how risk assessment outputs can be systematically translated into trust thresholds. This section presents example methodologies drawn from recent research and standardization efforts in the automotive domain, which serve to illustrate the practical application of risk-based RTL derivation rather than prescribing a singular solution.

Figure 8.2 illustrates the general flow from risk assessment to RTL threshold derivation. The process begins with a technical system model and defined operational scope, and proceeds through risk assessment to identify relevant threats and their associated risk levels, and culminates in the calculation of specific RTL threshold values for belief, disbelief, and uncertainty components.

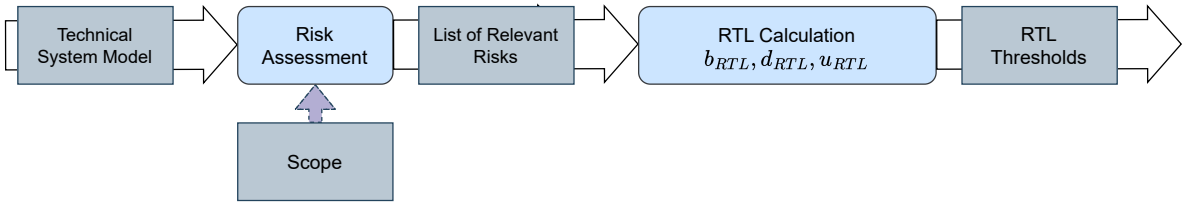

Figure 8.2: Risk-based RTL derivation flow in CASTOR. The process translates risk assessment outputs (attack feasibility and impact ratings) into concrete RTL threshold values.

8.4.1 Belief Component Calculation

One established approach for deriving the belief component of RTL is based on mapping risk levels from threat analysis directly to belief thresholds. This methodology, developed in the context of automotive cybersecurity, operates on the principle that higher risk scenarios necessitate stronger belief requirements.

The approach utilizes a belief threshold baseline (b_t) and a risk-dependent increment to compute the required belief level:

$$\Delta = \frac{1 - b_t}{5}, \quad 0 \leq b_t \leq 1 \tag{8.1}$$

$$b_{RTL} = b_t + ((R_{max} - 1) \times \Delta) \tag{8.2}$$

where

- b_{RTL} represents the minimum required belief
- R_{max} denotes the maximum risk level (on a scale of 1-5 derived from risk assessment)
- b_t is an organizational baseline that ensures minimum trust requirements even in low-risk scenarios
- Δ divides the available belief range (from b_t to 1) into five equal intervals corresponding to the five risk levels.

The baseline parameter b_t is determined based on organizational risk tolerance, regulatory requirements, or operational criticality considerations, establishing the minimum belief threshold for any scenario. Subsequently, R_{max} is identified by examining all threats relevant to the defined scope and selecting the highest risk level among them. In

Equation 8.2 ensures that as risk levels increase, the required belief threshold increases proportionally, with the baseline b_t providing a safety margin that prevents RTL from falling to zero, even for low-risk scenarios. The division by 5 in Equation 8.1 corresponds to the standardized five-level risk scale commonly employed in automotive risk assessment frameworks.

8.4.2 The Role of Attack Feasibility in Risk Assessment

A critical component of the risk level calculation is the attack feasibility assessment. In current practice, attack feasibility is manually determined by security analysts. This process involves:

- Identifying potential attack vectors and entry points
- Analyzing the steps required for an attacker to exploit vulnerabilities
- Evaluating the resources, expertise, and time required for successful exploitation
- Considering existing security controls and their effectiveness
- Constructing attack scenarios that chain multiple exploit steps

Example: A security analyst evaluating a network routing node subject to route injection attacks would examine: (1) whether the routing protocol implementation has known vulnerabilities (attack vector identification), (2) the complexity of crafting malicious routing advertisements (exploit steps), (3) the attacker's required knowledge of routing protocols and access to the network segment (resources and expertise), (4) the presence of route validation mechanisms such as cryptographic authentication or anomaly detection (existing controls), and (5) whether successful route injection enables subsequent attacks on data flowing through the compromised path (attack chaining). If the routing protocol lacks authentication, uses default configurations, and there are publicly available exploit tools, the analyst might assign a feasibility rating of "High". Combined with the severe impact of routing manipulation (affecting data integrity and availability across the network), this produces a high risk level ($R_{max} = 4$ or 5) that directly influences the calculation of the RTL through Equations 8.1 and 8.2.

This manual assessment produces a feasibility rating (typically on a scale from "very low" to "high") which, combined with impact ratings, yields the overall risk level R used in RTL calculation. While this approach is effective for design-time analysis of relatively static systems, it presents significant limitations for dynamic network environments.

8.4.3 Disbelief and Uncertainty Components

The RTL framework recognizes that trust assessment encompasses not only belief but also acceptable bounds on disbelief and uncertainty. Methodologies have been developed for calculating these components independently, each accounting for distinct aspects of system trustworthiness:

Disbelief thresholds are derived from impact analysis and residual risk considerations. The approach examines the potential consequences across multiple dimensions, that is, safety, economic, operational, and privacy impacts, and establishes maximum acceptable disbelief levels based on the severity of these impacts. Higher potential impacts necessitate lower disbelief tolerance. The calculation typically employs weighted impact ratings:

$$d_{RTL} = f(\text{weighted impact across safety, economic, operational, privacy}) \tag{8.3}$$

Where the function f inversely relates impact to acceptable disbelief, ensuring that high-impact scenarios demand near-zero tolerance for negative evidence. **Uncertainty thresholds** are determined by factors distinct from direct risk assessment, specifically:

- **Detectability:** The system's capability to identify misbehaviour or security incidents through monitoring mechanisms
- **Required assurance level:** The degree of confidence needed based on operational criticality and regulatory requirements

Systems with high detectability can tolerate greater uncertainty, as anomalies can be identified and addressed before they lead to failures. Conversely, safety-critical functions with low detectability require very low uncertainty thresholds to ensure adequate confidence despite limited observability.

These separate calculation methodologies for belief, disbelief and uncertainty reflect an important characteristic of RTL derivation: *each trust opinion component may be influenced by different risk factors and require distinct assessment approaches*. This raises the question of how to optimally balance these three dimensions when determining final RTL constraints, a question that remains an active area of research.

8.4.4 Limitations of Current Approaches and the CASTOR Gap

The methodologies presented above represent the state-of-the-art in risk-based RTL derivation for automotive and connected vehicle domains. However, they exhibit critical limitations when applied to the dynamic, topology-aware trust assessment scenarios that CASTOR addresses:

Manual Attack Feasibility Assessment

Current approaches rely on security analysts to manually construct attack scenarios and assess feasibility. This manual process:

- Cannot scale to large, dynamic network topologies with hundreds or thousands of potential attack paths
- Fails to account for runtime changes in network configuration, node availability, or vulnerability status
- May miss non-obvious multi-hop attack chains that emerge from the interaction of multiple low-severity vulnerabilities
- Provides only static, point-in-time assessments that do not reflect evolving threat landscapes

Lack of Cascading Risk Integration

The risk assessment methodologies underlying current RTL derivation typically evaluate threats in isolation or through predefined attack scenarios. They do not systematically account for:

- Cascading attacks where compromise of one node enables subsequent attacks on connected nodes
- Risk propagation through network paths, where a trusted path may traverse untrusted intermediate nodes
- The cumulative effect of multiple medium-risk nodes in an attack chain exceeding the risk of any individual node
- Dynamic changes in attack feasibility as network topology evolves during operation

8.4.5 Requirements for Advanced RTL Derivation in CASTOR

To address these limitations and enable effective trust-based routing in dynamic network environments, CASTOR must extend current RTL derivation methodologies in several key dimensions. The following requirements will be addressed in detail in Deliverable D4.2:

1. **Automatic Attack Path Identification**: Develop algorithms to automatically identify feasible attack paths in network topologies, considering multi-hop chains and evolving vulnerability landscapes without manual security analyst intervention.
2. **Temporal Attack Progression Modeling**: Integrate Markov Chain models to capture the temporal dynamics of attacks, representing how attackers transition between network states and accumulate capabilities through sequential compromises.
3. **Probabilistic Attack Feasibility Quantification**: Employ Monte Carlo simulation techniques to generate probabilistic assessments of attack feasibility under uncertainty, enabling ”what-if” scenario analysis for diverse threat models and defensive postures.
4. **Cascading Risk Aggregation**: Develop mechanisms to quantify cumulative risk along multi-hop paths, accounting for risk propagation and amplification effects that emerge from attack chaining.
5. **Multi-dimensional RTL Optimization**: Establish principled approaches for balancing trade-offs between belief, disbelief, and uncertainty thresholds when different calculation methodologies yield conflicting requirements, particularly in scenarios where strict belief requirements conflict with uncertainty tolerance needs.
6. **Dynamic RTL Adaptation**: Define mechanisms for updating RTL values in response to topology changes, newly discovered vulnerabilities, or evolving threat intelligence, while maintaining consistency with ongoing trust assessments.
7. **Topology-aware Risk Assessment**: Extend risk calculation to account for the structural properties of network graphs, recognizing that a node’s role in potential attack paths (e.g., as a gateway or pivoting point) influences its required trust level independent of its intrinsic vulnerability profile.

These requirements represent the core challenges that distinguish CASTOR’s approach to RTL derivation from existing methodologies. The detailed technical mechanisms for addressing each requirement, including specific algorithms for Markov Chain-based attack progression modeling and Monte Carlo-based feasibility quantification, will be elaborated in Deliverable D4.2.

8.4.6 Methodology Flexibility

It is important to emphasize that the equations and approaches presented in this section represent *one possible instantiation* of risk-based RTL derivation, drawn from automotive cybersecurity frameworks. The CASTOR architecture does not mandate any specific calculation methodology. Organizations may adapt these formulations, develop alternative mappings, or incorporate additional factors such as:

- Compliance requirements and regulatory constraints
- Operational considerations specific to network domains (e.g., industrial control systems vs. vehicular networks)
- Domain-specific threat models and attack taxonomies
- Organizational risk tolerance and security policies

The key requirement is that the chosen methodology produces quantifiable RTL values that can be consistently compared against runtime ATL assessments. This flexibility allows the framework to accommodate different operational domains, risk assessment standards, and organizational risk management practices while maintaining the fundamental principle of trust-based decision making through quantified threshold comparison.

8.5 Link with Trust Assessment

The RTL framework presented in this chapter is designed to integrate seamlessly with the runtime trust assessment mechanisms discussed in earlier chapters. This section briefly outlines how RTL thresholds interact with ATL computation to enable trust-based decision making in CASTOR.

8.5.1 Runtime Trust Decision Process

At runtime, the TAF continuously evaluates entities (nodes, data sources, communication channels) by collecting evidence from multiple trust sources and computing ATL opinions in the form of subjective logic triplets $(b_{ATL}, d_{ATL}, u_{ATL})$. The trust decision is made by comparing this runtime assessment against the design-time RTL thresholds:

$$\text{Trust Decision} = \begin{cases} \text{Accept} & \text{if } b_{ATL} \geq b_{RTL} \text{ AND } d_{ATL} \leq d_{RTL} \text{ AND } u_{ATL} \leq u_{RTL} \\ \text{Reject} & \text{otherwise} \end{cases} \tag{8.4}$$

All three constraints must be satisfied simultaneously for an entity to be deemed trustworthy. If any single threshold is violated- insufficient belief, excessive disbelief, or unacceptable uncertainty- the trust decision fails, triggering alternative actions such as rejecting data, avoiding routing paths through the untrusted node, or escalating to human operators for manual intervention.

8.5.2 Integration with Trust-Aware Routing

In the context of CASTOR's trust-aware routing mechanisms, RTL plays a critical role in path selection decisions. When evaluating candidate routing paths, each intermediate node along a potential path must satisfy its corresponding RTL requirements. The path-level trust assessment aggregates individual node

ATL values using subjective logic conjunction. For a path to be deemed trustworthy, each intermediate node must satisfy its individual RTL requirements, and the aggregated path-level trust must reflect the cumulative trustworthiness of all constituent elements.

This integration creates a feedback loop between risk assessment and trust assessment:

- **Design-time**: Risk assessment derives RTL thresholds based on threat analysis and impact evaluation.
- **Runtime**: Trust assessment monitors evidence, computes ATL, and compares against RTL.
- **Decision**: Routing mechanisms accept or reject paths based on trust decisions, effectively operationalizing risk-based security requirements.
- **Adaptation**: Observed attack patterns or trust violations feed back into risk models, potentially triggering RTL recalculation for enhanced security posture.

8.5.3 Handling Trust Decision Failures

When an entity fails to meet RTL requirements (ATL $\not\geq$ RTL), CASTOR must respond appropriately based on the operational context:

- **Data validation**: Reject messages or data originating from untrusted sources, preventing potentially compromised information from influencing system behavior.
- **Path avoidance**: Exclude untrusted nodes from routing path selection, even if they offer shorter or lower-latency routes, prioritizing security over performance.
- **Degraded service**: Accept data or utilize paths with reduced functionality, such as limiting bandwidth, applying additional verification steps, or restricting access to sensitive operations.
- **Alerting and logging**: Generate security events for monitoring systems, enabling detection of emerging threats or systematic trust degradation across network segments.
- **Re-assessment**: Trigger additional evidence collection or invoke alternative trust sources to obtain higher-confidence ATL assessments before making final decisions.

The specific response strategy depends on the application requirements, the severity of the trust violation (e.g., marginal vs. significant RTL failure), and the availability of alternative trusted resources.

8.5.4 Evolution of RTL in Operational Systems

While RTL is primarily a design-time artifact, it is not entirely static. As systems evolve—through software updates, topology changes, discovery of new vulnerabilities, or shifts in threat landscapes—RTL values may require recalculation to maintain alignment between design-time risk assessments and runtime security requirements. The mechanisms for dynamic RTL adaptation, including triggers for re-assessment and methods for ensuring consistency during transitions, will be addressed in Deliverable D4.2 as part of the advanced RTL derivation framework.

8.5.5 Summary

RTL serves as the critical bridge between design-time risk analysis and runtime trust-based decision making. By establishing quantifiable thresholds derived from systematic risk assessment, RTL enables CASTOR to operationalize security requirements in a measurable, consistent, and auditable manner. The trust decision mechanism (Equation 8.4) provides a clear, objective criterion for accepting or rejecting entities based on their demonstrated trustworthiness, ensuring that the dynamic trust assessment framework remains grounded in rigorous risk-based foundations.

Chapter 9

Optimization

9.1 Optimization Vocabulary

- **Trusted Path Routing**: Routing approach that selects end-to-end network paths by jointly optimizing network performance and trust-related constraints.
- **Network Graph**: Directed graph representation of the network, where nodes correspond to devices and edges to communication links.
- **Feasible Path**: A loop-free path between a source and destination that satisfies all policy, trust, and performance constraints.
- **Objective Vector**: A vector of network- and trust-related metrics used to evaluate and compare candidate paths.
- **Multi-objective Optimization**: Optimization framework that simultaneously considers multiple, potentially conflicting objectives.
- **Trust Constraint**: Requirement enforcing that a path-level trust metric exceeds a predefined minimum threshold.
- **Forbidden Nodes or Links**: Network elements excluded from routing due to security, policy, or regulatory restrictions.
- **Pareto Dominance**: A solution dominates another if it is no worse in all objectives and strictly better in at least one.
- **Pareto Front**: The set of all non-dominated solutions representing optimal trade-offs among objectives.
- **Network-related Attributes**: Scalar metrics such as latency, bandwidth, availability, and resource utilization.
- **Trust-related Attributes**: Metrics expressing confidence in the secure, reliable, and policy-compliant behavior of network elements.
- **Subjective Logic Opinion**: Trust representation defined by belief, disbelief, uncertainty, and base rate.
- **Projected Probability**: Scalar value derived from a Subjective Logic opinion, combining belief and uncertainty for optimization.

- **Path-level Aggregation**: Process of combining node- and edge-level attributes into path-level metrics.
- **Multi-objective Dijkstra Algorithm**: Exact label-setting algorithm that computes all Pareto-optimal paths in multi-objective shortest path problems.
- **Dimensionality Reduction**: Reduction of the number of optimization objectives by aggregating attributes to improve scalability.
- **Bi-objective Optimization**: Optimization formulation balancing one aggregated network objective and one aggregated trust objective.
- **Quantum Annealing**: Optimization paradigm that seeks the minimum of an Ising Hamiltonian using quantum-mechanical effects.
- **Quantum-inspired Optimization**: Classical algorithms that emulate quantum or physical dynamics while running on conventional hardware.
- **Simulated Bifurcation**: Hamiltonian-based classical algorithm exploiting bifurcation dynamics to solve Ising-type problems.
- **Ballistic Simulated Bifurcation**: Nonadiabatic variant of Simulated Bifurcation using momentum-driven dynamics for faster convergence.
- **Discrete Simulated Bifurcation**: Simulated Bifurcation variant that discretizes interaction terms to improve robustness.
- **Ising Hamiltonian**: Energy function defining interactions between binary spin variables, whose minimum encodes the solution.
- **Quadratic Unconstrained Binary Optimization (QUBO)**: Binary optimization framework using quadratic cost functions compatible with Ising-based solvers.
- **Scalarization**: Technique converting multiple objectives into a single objective, typically via weighted combinations.
- **Augmented Lagrangian Method**: Constraint-handling approach combining penalty terms and Lagrange multipliers to improve feasibility.

9.2 Problem formulation

Trusted Path Routing addresses the problem of selecting end-to-end communication paths that jointly satisfy network performance requirements and trust-related constraints within the CASTOR architecture. In contrast to traditional routing approaches that rely solely on network-centric metrics, trusted path routing treats trust as an integral dimension of the routing decision. The objective is therefore to identify paths that balance performance, reliability, and trustworthiness, in accordance with service requirements and security policies introduced in earlier sections of this deliverable.

Let the network be represented by a directed graph

$$G = (V, E), \tag{9.1}$$

where the vertices V denote the set of nodes and the edges E the communication links. For a given source–destination pair $(s, t) \in V \times V$, let

$$\Pi(s, t) \tag{9.2}$$

denote the set of all feasible paths connecting s to t.

Each path $\pi \in \Pi(s,t)$ is associated with a vector of objective values

$$f(\pi) = \big(f_1(\pi), f_2(\pi), \ldots, f_m(\pi)\big), \tag{9.3}$$

where each component $f_i(\pi)$ represents an aggregated network- or trust-related attribute characterizing the path. These attributes capture performance, reliability, and trust properties, and are derived from node- and edge-level metrics defined on the underlying graph.

The trusted path routing problem is thus formulated as a multi-objective optimization problem:

$$\begin{aligned} \min_{\pi \in \Pi(s,t)} \quad & f(\pi) \\ \text{subject to} \quad & \pi \in \Pi(s,t), \\ & T_i(\pi) \geq \tau_i, \quad i = 1, \ldots, N, \\ & \pi \cap V_{\text{forbidden}} = \varnothing, \\ & g_j(\pi) \leq c_j, \quad j = 1, \ldots, J. \end{aligned} \tag{9.4}$$

The constraint $\pi \in \Pi(s,t)$ enforces path feasibility, ensuring that the selected solution corresponds to a valid, loop-free path connecting the source node s to the destination node t in the network graph. The constraints $T_i(\pi) \geq \tau_i$, for $i = 1, \ldots, N$, represent trust-related requirements. Each function $T_i(\pi)$ denotes a path-level trust measure associated with a specific trust property, such as integrity, confidentiality, or reliability. The corresponding threshold τ_i defines the minimum acceptable level for that property, allowing multiple independent trust dimensions to be enforced simultaneously.

The constraint $\pi \cap V_{\text{forbidden}} = \varnothing$ captures policy and security restrictions, excluding nodes or links that are disallowed due to administrative policies, regulatory requirements, or security considerations. Finally, the constraints $g_j(\pi) \leq c_j$, for $j = 1, \ldots, J$, impose network performance or resource bounds. These constraints limit aggregated path-level metrics such as latency, bandwidth consumption, or resource utilization, ensuring that selected paths satisfy predefined quality-of-service or capacity requirements.

The formulation above naturally leads to a multi-objective optimization setting, as the objective vector

$$f(\pi) = \big(f_1(\pi), \ldots, f_m(\pi)\big) \tag{9.5}$$

generally comprises multiple, potentially conflicting network- and trust-related criteria. Contrary to single-objective optimization, where a unique optimal solution can typically be identified, multi-objective optimization admits no single best solution. Instead, optimality is defined in terms of Pareto dominance[45] , yielding a set of trade-off solutions.

Let

$$Y = \big\{ f(\pi) \mid \pi \in \Pi(s,t) \big\} \tag{9.6}$$

denote the set of attainable objective vectors. A path $\pi_1 \in \Pi(s,t)$ is said to dominate another path $\pi_2 \in \Pi(s,t)$ if it is no worse in all objectives and strictly better in at least one, i.e.,

$$f(\pi_1) \succ f(\pi_2). \tag{9.7}$$

The Pareto front [45] (also referred to as the Pareto frontier or Pareto curve) is defined as the set of all non-dominated objective vectors in Y, corresponding to Pareto-optimal paths. Accordingly, the output of the Optimization Engine is not a single path but a set of Pareto-optimal or near-optimal paths, each representing a different trade-off between network performance and trust-related objectives.

In this context, a feasible solution corresponds to any path $\pi \in \Pi(s,t)$ satisfying all imposed constraints. A feasible path is considered dominated if there exists another feasible path whose objective vector dominates it, while a non-dominated path represents a Pareto-optimal solution. The collection of such non-dominated paths therefore characterizes the trade-offs inherent in the trusted path routing problem and supports the selection of ranked alternatives, such as primary and backup paths.

9.2.1 Network and Trust Attributes

The Optimization Engine operates on the above network abstraction as a multi-weighted graph, where both nodes $v \in V$ and edges $e \in E$ are annotated with multiple attributes. These attributes include network-related metrics, such as performance, availability, and resource status, as well as trust-related attributes provided by the Trust Assessment Framework (TAF). This unified representation enables trusted path routing to be addressed systematically as a multi-objective optimization task, in which network efficiency and trustworthiness are optimized simultaneously. The nature and representation of these heterogeneous attributes, which capture complementary aspects of system behavior and reflect the dual nature of the routing problem, are described in the following section.

Network-related attributes. Network-related attributes are modeled as numerical scalar metrics that characterize the performance, availability, and resource status of nodes and links. Representative examples include latency, hop count, bandwidth, packet loss, availability, and utilization-related metrics. Network-related attributes are defined at the level of nodes and links and represent performance-, availability-, and resource-related properties.

The manner in which these attributes are aggregated along candidate paths is not specified at this stage and is described in detail in Section 9.2.4. Rather than enforcing a fixed aggregation rule, the Optimization Engine supports configurable aggregation functions, allowing network metrics to be interpreted and combined according to the requirements expressed in each path profile.

Trust-related attributes. Trust-related attributes capture the degree of confidence that nodes and links behave as expected with respect to security, reliability, and policy compliance. In accordance with the trust model defined in Chapter 4 of this deliverable, trust attributes are represented using Subjective Logic opinions [26]

$$\omega = (b, d, u, a), \tag{9.8}$$

where belief b, disbelief d, uncertainty u, and base rate a jointly encode available evidence and epistemic uncertainty. These opinions may be binomial or multinomial, enabling the representation of both simple trust propositions and more complex trust assessments.

Trust opinions are produced and continuously updated by the Trust Assessment Framework (TAF), which integrates evidence from multiple sources and reflects the dynamic nature of trust in operational environments. At this stage, trust attributes are treated abstractly as path-relevant quantities; their projection to scalar values and their aggregation along paths are described in subsequent sections.

9.2.2 From Opinions to Projected Probabilities

Subjective Logic provides a principled framework for representing trust by explicitly modeling belief, uncertainty, and base rate. However, the optimization procedures employed by the Optimization Engine operate on scalar quantities and require numerical representations in order to evaluate, compare, and rank candidate solutions. To bridge this gap, trust opinions associated with nodes and edges are mapped to scalar values through the use of projected probabilities.

Given a Subjective Logic opinion

$$\omega = (b, d, u, a), \tag{9.9}$$

the projected probability is defined as [26]

$$P(\omega) = b + a \cdot u. \tag{9.10}$$

This quantity corresponds to the expected probability that the underlying trust proposition holds, conditioned on the available evidence and the base rate. The contribution of uncertainty is explicitly retained through the term $a \cdot u$, ensuring that incomplete or ambiguous evidence is neither ignored nor overemphasized.

The adoption of projected probabilities constitutes a controlled abstraction that preserves the decision-relevant semantics of trust while enabling its integration into numerical optimization frameworks. By expressing trust attributes as scalar values, they can be aggregated along candidate paths and combined with network-related metrics within a unified multi-objective optimization formulation. This mapping therefore provides the essential interface between the trust representation mechanisms introduced earlier and the optimization methodology developed in the subsequent sections.

9.2.3 Path-level Composition of Trust and Network Attributes

Path evaluation in the Optimization Engine is based on the aggregation of network and trust attributes along candidate paths. Given a candidate path

$$\pi = (v_0, e_1, v_1, \ldots, e_n, v_n), \tag{9.11}$$

path-level attributes are derived by combining the node- and edge-level metrics associated with the elements of the path.

Network-related attributes are aggregated along candidate paths according to aggregation functions specified by the selected path profile. The choice of aggregation function depends on the semantics of each metric and may include additive aggregation (e.g., latency), multiplicative aggregation (e.g., availability), or extremal operators (e.g., minimum residual bandwidth). This flexible aggregation framework enables heterogeneous network metrics, defined at the level of nodes and links, to be consistently composed into meaningful path-level quantities for optimization.

In contrast, trust-related attributes are composed using the Subjective Logic conjunction (multiplication) operator, which enables reasoning about the joint satisfaction of multiple trust conditions. At the optimization level, the corresponding projected probabilities are combined multiplicatively to obtain a path-level trust value:

$$P(\pi) = \prod_{x \in \pi} P(\omega_x), \tag{9.12}$$

where ω_x denotes the trust opinion associated with node or edge x. This formulation reflects the requirement that a path is considered trustworthy only if all of its constituent elements satisfy the required trust properties.

The resulting aggregated network and trust attributes provide a unified quantitative characterization of each candidate path and form the basis for the multi-objective optimization process.

9.3 Methodologies analysis

This section describes the algorithmic methodologies adopted to solve the trusted path routing problem formulated in the previous sections. Within the CASTOR project, the use of Quantum Annealing (QA) [39] was initially proposed as a promising approach for addressing the underlying combinatorial optimization problem through QUBO formulations [38] . However, current limitations of quantum hardware, particularly in terms of qubit count and connectivity, do not yet allow the practical deployment of QA for networks of realistic scale [37, 32, 1]. To address these limitations, CASTOR adopts a hybrid algorithmic strategy that complements existing routing technologies and control-plane mechanisms. Current routing frameworks, such as Segment Routing (SR) and Flex-Algo, do not natively support true multi-objective shortest path

computation, but instead rely on predefined cost functions, constraints, and multiple algorithm instances. In contrast, CASTOR introduces state of the art exact classical multi-objective shortest path algorithms [13, 42] to explicitly address the full multi-objective routing problem. The ability to deploy these algorithms in real time or in a planning context is determined by the scalability and performance of the underlying optimization engine, rather than by limitations of the routing model itself. In parallel, CASTOR explores quantum-inspired optimization techniques [48] as scalable alternatives in order to preserve the conceptual framework introduced during the proposal phase, namely the use of QUBO and Ising formulations originally envisaged for Quantum Annealing. These methods operate on classical hardware while retaining annealing-inspired dynamics and Hamiltonian-based problem representations, allowing CASTOR to maintain the intended optimization paradigm despite current quantum hardware constraints. In addition, physics-inspired classical heuristic algorithms, such as Simulated Annealing [3], are considered as standard stochastic solvers for Ising-type formulations, providing a well-established reference for evaluating the behavior and performance of quantum-inspired dynamics on equivalent problem representations.[

9.3.1 Exact algorithms

Multi-objective Dijkstra

The Multi-Objective Dijkstra Algorithm (MOD) [13], also referred to as the Multiobjective Dijkstra Algorithm (MDA), is an exact label-setting algorithm for the Multiobjective Shortest Path (MOSP) problem. It generalizes the classical Dijkstra algorithm [17]to the case where multiple, potentially conflicting objectives are optimized simultaneously and aims to compute a minimum complete set of Pareto-optimal paths between a source node and all reachable nodes in the network.

In contrast to single-objective shortest path algorithms, MOD associates each node with a set of labels, where each label represents a distinct non-dominated cost vector corresponding to a feasible path from the source to that node. A label encodes both the accumulated objective values and a reference to its predecessor, allowing efficient path reconstruction while avoiding explicit storage of full paths. Dominance relations between labels are evaluated using the Pareto order, and dominated labels are discarded to prune the search space.

The algorithm follows a label-setting strategy. At each iteration, a lexicographically smallest tentative label is extracted from a priority queue and made permanent, meaning it is guaranteed to be non-dominated. This property relies on the principle of optimality, which holds under the assumption of non-negative arc costs and ensures that efficient paths are composed of efficient subpaths. Once a label becomes permanent, it is extended along outgoing edges to generate new candidate labels, which are then filtered through dominance checks before being considered for further expansion.

A key distinguishing feature of the MDA is that at most one tentative label per node is stored in the priority queue at any time, bounding the queue size by the number of nodes in the graph. This design choice significantly reduces memory usage and allows the algorithm to achieve an output-sensitive complexity, where the running time depends not only on the network size but also on the cardinality of the Pareto front. While the worst-case complexity grows with the number of objectives and the number of efficient paths per node, the algorithm is provably more efficient than classical label-setting approaches such as Martins' algorithm and has demonstrated substantial performance gains in practice.

The computational complexity of Dijkstra-based shortest path algorithms depend strongly on the number of optimization objectives d. Table 9.1 summarizes the asymptotic running time of the classical Dijkstra algorithm, the bi-objective case, and the general multi-objective case, as reported in Maristany de las Casas et al. (2021).

Here, n is the number of nodes in the network graph, m the number of edges in the network graph, d the number of objective functions (optimization criteria), N the total number of Pareto-optimal (efficient)

Table 9.1: Running time of Dijkstra-based algorithms depending on the number of objectives

Number of objectives d	Running time (output-sensitive form)
$d = 1$	$\mathcal{O}(n \log n + m)$
$d = 2$	$\mathcal{O}(N \log n + N_{\max} m)$
$d \geq 3$	$\mathcal{O}(d\,(N \log n + N_{\max}^2 m))$

paths generated by the algorithm, and $N_{\max}$ the maximum number of Pareto-optimal paths associated with any single node.

In the context of CASTOR, the multi-objective Dijkstra algorithm serves as the baseline exact method for computing trusted paths and extracting the full Pareto front. It provides a reference against which reduced-dimensional formulations and quantum-inspired optimization methods are evaluated. However, as the number of objectives increases, the size of the Pareto front may grow rapidly, motivating the use of dimensionality reduction strategies and alternative optimization approaches discussed in the following sections.

9.3.2 Dimensionality Reduction via Attribute Conjunction

The complexity results summarized in Table 9.1 illustrate the strong dependence of Dijkstra-based multi-objective shortest path algorithms on the number of optimization criteria. While the single-objective case admits a polynomial-time solution, the introduction of additional objectives leads to a rapid increase in the number of efficient paths that must be maintained during the search. As shown in the table, the bi-objective case remains output-sensitive, with computational cost scaling proportionally with the size of the Pareto front. In contrast, when three or more objectives are considered, the complexity exhibits a quadratic dependence on the maximum number of efficient paths per node, reflecting the increasing cost of dominance checks and label management. These observations provide a clear algorithmic justification for restricting the optimization to two objectives whenever possible, motivating the adoption of a bi-objective formulation in the following sections.

To reduce the computational complexity of the optimization problem, a dimensionality reduction step can be applied prior to path computation. For trust-related attributes, dimensionality reduction is achieved through the use of the Subjective Logic conjunction operator across multiple trust opinions associated with the same node or edge. For example, separate opinions expressing trust in integrity and confidentiality can be conjuncted into a single composite opinion representing the joint satisfaction of both properties. This composite opinion preserves the semantics of trust under uncertainty and is subsequently projected onto a scalar probability, yielding a single trust value suitable for optimization.

In contrast, network-related attributes generally do not admit conjunction-based composition, as they represent heterogeneous performance metrics with distinct physical interpretations. Instead, alternative aggregation strategies can be employed to reduce dimensionality, such as weighted combinations, profile-specific cost functions, or policy-driven scalarization. These approaches allow multiple network metrics to be collapsed into a single aggregated network objective while retaining their relative importance as defined by service requirements.

Following these reductions, the original high-dimensional multi-objective optimization problem can be reformulated as a bi-objective optimization problem, typically balancing one aggregated network objective against one aggregated trust objective. As discussed in [13], bi-objective formulations significantly improve algorithmic scalability. In particular, they limit the growth of the Pareto front and enable the use of more efficient, output-sensitive shortest path algorithms. This makes bi-objective optimization especially well suited for large-scale and dynamic network environments, where exact high-dimensional multi-objective methods may become computationally prohibitive.

9.3.3 Quantum- and Physics-Inspired algorithms

Quantum Annealing (QA) is an optimization paradigm that seeks the ground state of an Ising Hamiltonian by exploiting fundamental quantum-mechanical effects. Its evolution relies on quantum superposition, which enables the simultaneous exploration of multiple configurations, and quantum tunneling, which allows the system to transition through energy barriers rather than over them. These mechanisms can be particularly effective in escaping narrow or deep local minima that often hinder classical optimization methods. As such, QA provides a natural computational framework for solving combinatorial optimization problems formulated in QUBO or Ising form. Despite these conceptual advantages, the practical deployment of QA within the CASTOR project is currently constrained by the limitations of available quantum hardware. In particular, the restricted number of qubits, limited connectivity, noise, and embedding overhead prevent the direct application of QA to network optimization problems of realistic scale and complexity [39]. These hardware constraints significantly limit the size of problem instances that can be addressed and motivate the exploration of alternative approaches. To overcome these limitations while retaining the mathematical structure and physical intuition of QA, CASTOR investigates quantum-inspired optimization techniques. These methods map combinatorial optimization problems onto physical systems whose low-energy states correspond to optimal or near-optimal solutions, but are executed entirely on classical hardware, overcoming scalability issues. Typical examples include quantum annealing–inspired algorithms [48] that simulate the dynamics of quantum or analog Ising machines [33], such as simulated coherent Ising machines (SimCIM) and related nonlinear oscillator–based approaches. These techniques relax discrete variables into continuous ones and employ annealing-like or dynamical evolution schemes to efficiently explore complex energy landscapes. Within this family, Simulated Bifurcation (SB) [24] emerges as a particularly relevant approach. SB is a Hamiltonian-based, fully classical algorithm that exploits bifurcation phenomena in nonlinear dynamical systems to solve Ising-type optimization problems. By closely emulating the adiabatic dynamics underlying quantum annealing, while avoiding the constraints of quantum hardware, SB provides a scalable and practical alternative for QUBO-based optimization and forms the primary quantum-inspired methodology investigated in this work.

9.3.4 Simulated Bifurcation – General Description

Simulated Bifurcation (SB) is a quantum-inspired optimization algorithm derived from the classical simulation of adiabatic evolutions in nonlinear Hamiltonian systems exhibiting bifurcation phenomena. The method is inspired by quantum adiabatic optimization with nonlinear oscillators, but it operates entirely within a classical-mechanical framework. Continuous dynamical variables evolve under a time-dependent Hamiltonian, and bifurcations in the system drive the variables toward discrete states corresponding to Ising spins. The final solution is obtained by taking the sign of these variables, which encodes a low-energy configuration of the target Ising Hamiltonian.

Depending on how the underlying dynamical evolution is implemented and controlled, SB admits several algorithmic variants that trade off adiabaticity, convergence speed, and robustness. In particular, three main flavors are commonly considered: adiabatic Simulated Bifurcation (aSB) [24] , ballistic Simulated Bifurcation (bSB), and discrete Simulated Bifurcation (dSB) [23]. In the context of the CASTOR project, the focus is placed on bSB and dSB, as these variants have been shown to exhibit increased robustness and faster convergence compared to the adiabatic formulation, making them more suitable for large-scale and practical optimization scenarios.

Definition of Simulated Bifurcation parameters. The dynamics of both ballistic and discrete Simulated Bifurcation are governed by three global parameters: a_0, $a(t)$, and c_0.

The parameter $a(t)$ is a time-dependent control parameter that is increased from zero during the evolution. Its role is to induce bifurcations in the system by destabilizing the trivial equilibrium at $x_i = 0$,

thereby driving the dynamical variables toward the discrete attractors corresponding to Ising spin states.

The parameter $a_0 > 0$ is a constant that sets the characteristic time scale of the Hamiltonian dynamics and determines the coupling between the position variables x_i and their conjugate momenta y_i. In the implementation considered in this work, a_0 is fixed to a constant value.

The parameter $c_0 > 0$ scales the interaction term and encodes the strength of the Ising couplings into the Simulated Bifurcation dynamics. It determines the relative influence of the coupling matrix J_{ij} on the system evolution.

Ballistic Simulated Bifurcation (bSB). Ballistic Simulated Bifurcation is a nonadiabatic variant designed to improve convergence speed and solution quality. Instead of slowly tracking bifurcating minima, the system undergoes rapid, momentum-driven dynamics that push variables toward discrete boundaries. Perfectly inelastic constraints force variables to settle at their binary limits, ensuring convergence to stable local minima of the Ising energy.

Hamiltonian formulation. The Hamiltonian of the ballistic SB system is defined as

$$H_{\mathrm{bSB}} = \frac{a_0}{2} \sum_{i=1}^{N} y_i^2 + V_{\mathrm{bSB}}, \tag{9.13}$$

with the potential term

$$V_{\mathrm{bSB}} = -\frac{a_0 - a(t)}{2} \sum_{i=1}^{N} x_i^2 - \frac{c_0}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} J_{i,j}\, x_i x_j, \quad \text{for } |x_i| \leq 1. \tag{9.14}$$

Outside this domain, the potential is defined as

$$V_{\mathrm{bSB}} = \infty, \tag{9.15}$$

corresponding to perfectly inelastic walls at $x_i = \pm 1$.

Continuous-time dynamics. For each spin $i = 1, \ldots, N$, the equations of motion are given by

$$\dot{x}_i = a_0\, y_i, \tag{9.16}$$

$$\dot{y}_i = -\big[a_0 - a(t)\big] x_i + c_0 \sum_{j=1}^{N} J_{i,j}\, x_j. \tag{9.17}$$

Discrete-time update (symplectic Euler). Using a symplectic Euler discretization with time step Δt, the update equations read

$$y_i^{(k+1)} = y_i^{(k)} + \left(-\big[a_0 - a(t_k)\big] x_i^{(k)} + c_0 \sum_{j=1}^{N} J_{i,j}\, x_j^{(k)} \right) \Delta t, \tag{9.18}$$

$$x_i^{(k+1)} = x_i^{(k)} + a_0\, y_i^{(k+1)}\, \Delta t. \tag{9.19}$$

If $|x_i^{(k+1)}| > 1$, perfectly inelastic boundary conditions are enforced:

$$x_i^{(k+1)} \leftarrow \mathrm{sgn}\left(x_i^{(k+1)}\right), \qquad y_i^{(k+1)} \leftarrow 0. \tag{9.20}$$

Discrete Simulated Bifurcation (dSB). Discrete Simulated Bifurcation is an enhanced SB variant designed to suppress analog errors and improve solution accuracy. In dSB, continuous interaction terms are partially discretized by replacing neighboring dynamical variables with their signs when computing coupling forces. This modification introduces discontinuities that enable tunneling-like transitions in the classical dynamics, allowing the system to escape local minima while preserving the parallel nature of the SB framework.

Hamiltonian formulation. The Hamiltonian of the discrete SB system is defined as

$$H_{\mathrm{dSB}} = \frac{a_0}{2} \sum_{i=1}^{N} y_i^2 + V_{\mathrm{dSB}}, \tag{9.21}$$

with

$$V_{\mathrm{dSB}} = -\frac{a_0 - a(t)}{2} \sum_{i=1}^{N} x_i^2 - c_0 \sum_{i=1}^{N} \sum_{j=1}^{N} J_{i,j}\, x_i\, \mathrm{sgn}(x_j), \quad \text{for } |x_i| \leq 1, \tag{9.22}$$

and

$$V_{\mathrm{dSB}} = \infty \tag{9.23}$$

otherwise.

Continuous-time dynamics. The equations of motion for dSB are given by

$$\dot{x}_i = a_0\, y_i, \tag{9.24}$$

$$\dot{y}_i = -\big[a_0 - a(t)\big] x_i + c_0 \sum_{j=1}^{N} J_{i,j}\, \mathrm{sgn}(x_j). \tag{9.25}$$

Discrete-time update. The corresponding discrete-time updates are

$$y_i^{(k+1)} = y_i^{(k)} + \left(-\big[a_0 - a(t_k)\big] x_i^{(k)} + c_0 \sum_{j=1}^{N} J_{i,j}\, \mathrm{sgn}\Big(x_j^{(k)}\Big) \right) \Delta t, \tag{9.26}$$

$$x_i^{(k+1)} = x_i^{(k)} + a_0\, y_i^{(k+1)}\, \Delta t. \tag{9.27}$$

As in the ballistic case, boundary conditions are enforced whenever $|x_i| > 1$:

$$x_i \leftarrow \mathrm{sgn}(x_i), \qquad y_i \leftarrow 0. \tag{9.28}$$

Extraction of the spin configuration. Both ballistic and discrete Simulated Bifurcation operate on continuous dynamical variables $x_i(t) \in [-1, 1]$, whose evolution is governed by Hamiltonian dynamics and subject to inelastic boundary constraints. The design of the SB potential ensures that stable attractors of the dynamics are located at the boundaries $x_i = \pm 1$, which correspond to discrete Ising spin states.

Upon termination of the dynamical evolution, the final Ising spin configuration is obtained through a deterministic projection given by

$$s_i = \mathrm{sgn}(x_i), \tag{9.29}$$

where $s_i \in \{-1, +1\}$ denotes the Ising spin associated with variable i. This projection maps the continuous SB state to a discrete spin configuration that defines a candidate solution of the Ising Hamiltonian introduced in Section 9.3.6.

In ballistic Simulated Bifurcation, the inelastic boundary conditions enforce $|x_i| = 1$ at convergence, guaranteeing that the extracted spin configuration corresponds to a stable local minimum of the Ising energy. In discrete Simulated Bifurcation, the use of sign-discretized interaction terms further promotes transitions between basins of attraction, enabling the system to escape shallow local minima while preserving the same spin-extraction rule.

9.3.5 Simulated Annealing: General Description and Operating Principle

Simulated Annealing (SA) is a classical stochastic optimization algorithm inspired by the physical process of thermal annealing, in which a material is slowly cooled to reach a low-energy crystalline state. In the context of combinatorial optimization, SA searches for low-energy configurations of an objective function by introducing controlled randomness that allows the algorithm to escape local minima during the optimization process.

The algorithm operates by iteratively proposing random modifications to the current solution and evaluating the resulting change in the objective function. If the proposed move leads to a lower energy state, it is accepted deterministically. If the move increases the energy, it is accepted with a probability that depends on both the energy increase and a control parameter known as the temperature. This probabilistic acceptance mechanism enables the exploration of the solution space beyond greedy descent and prevents premature convergence to suboptimal local minima.

The temperature is gradually reduced according to a predefined annealing schedule, shifting the algorithm from an exploratory regime at high temperature to a more exploitative regime at low temperature. As the temperature approaches zero, the algorithm increasingly favors energy-decreasing moves, effectively converging toward a locally or globally optimal solution. When applied to Ising or QUBO formulations, SA typically operates directly on discrete spin variables, making it a widely used baseline method for solving Ising-type optimization problems on classical hardware.

9.3.6 Quadratic Unconstrained Binary Optimization and Ising Formulation

To provide the mathematical foundation for the quantum-inspired optimization algorithms considered in this work, the Quadratic Unconstrained Binary Optimization (QUBO) and Ising formulations are introduced first. QUBO is a general mathematical framework for formulating combinatorial optimization problems using binary decision variables [22]. In the QUBO formulation, the objective is to minimize a quadratic cost function that captures both individual variable contributions and pairwise interactions between variables. All constraints of the original problem are incorporated directly into the objective function through appropriately weighted quadratic penalty terms, thereby transforming a constrained optimization problem into an unconstrained one. This property makes QUBO particularly attractive, as it provides a unified representation compatible with both quantum annealers and quantum-inspired optimization algorithms.

A QUBO problem is defined as the minimization of a quadratic form

$$\min_{x \in \{0,1\}^N} \; x^\top Q x, \tag{9.30}$$

where $x = (x_1, \ldots, x_N)^\top$ is a vector of binary decision variables and Q is a symmetric matrix encoding linear costs along its diagonal elements Q_{ii} and quadratic interactions in its off-diagonal elements Q_{ij} for $i \neq j$. Constraints are enforced by introducing penalty terms into Q that assign higher energy values to infeasible configurations. As a result, feasible solutions correspond to low-energy states of the objective function, and the global minimum of the QUBO formulation yields an optimal feasible solution to the original combinatorial problem.

An alternative but mathematically equivalent formulation is provided by the Ising model [31], which originates from statistical physics and describes systems of interacting spins. In the Ising representation, binary decision variables are expressed as spin variables taking values in $\{-1, +1\}$, and the optimization task corresponds to finding the spin configuration that minimizes the system's energy. This formulation is particularly natural for quantum annealing and quantum-inspired optimization methods, as it directly maps the problem onto an energy landscape whose ground state represents the optimal solution.

An Ising optimization problem is defined through the Hamiltonian

$$H(s) = \sum_{i=1}^{N} h_i \, s_i + \sum_{i<j} J_{ij} \, s_i s_j, \tag{9.31}$$

where $s = (s_1, \ldots, s_N)^\top$ with $s_i \in \{-1, +1\}$. The coefficients h_i represent local fields acting on individual spins, while J_{ij} denote pairwise couplings between spins. The objective is to determine the spin configuration that minimizes the Hamiltonian.

Transformation between binary and spin variables. The QUBO and Ising formulations are directly related through a linear variable transformation that maps binary variables $x_i \in \{0, 1\}$ to spin variables $s_i \in \{-1, +1\}$. A commonly used transformation is

$$s_i = 2x_i - 1, \qquad x_i = \frac{1}{2}(s_i + 1). \tag{9.32}$$

Under this transformation, both formulations describe the same optimization landscape, and minimizing the Ising Hamiltonian is equivalent to solving the corresponding QUBO problem. This equivalence allows optimization problems to be expressed interchangeably in QUBO or Ising form, providing a common mathematical foundation for quantum annealing, quantum-inspired, and physics-based optimization algorithms.

9.3.7 QUBO for Multi-objective Optimization

Although, QUBO and Ising formulations provide a powerful and unified framework for combinatorial optimization, they are inherently designed to handle single-objective optimization problems, where the goal is to minimize a single scalar energy function. In multi-objective optimization problems, however, the objective is to simultaneously optimize multiple, often conflicting, criteria, resulting in a set of trade-off solutions rather than a single optimum. This fundamental mismatch poses a challenge when directly applying QUBO-based solvers to multi-objective problems.

A common approach to addressing multi-objective optimization within the QUBO framework is to repeatedly solve single-objective QUBO instances obtained through scalarization of the individual objectives, such as weighted-sum formulations [11]. While this strategy is conceptually straightforward, it is well known that simple scalarization techniques may fail to adequately capture the full Pareto front, particularly when the Pareto front is nonconvex. As a result, naive scalarization-based approaches may yield incomplete or biased representations of the trade-offs among objectives.

However, given the central role of the QUBO formulation in quantum annealing, recent work has shown that this limitation can be mitigated through carefully designed scalarization and decomposition strategies [25]. These approaches enable the systematic exploration of nonconvex Pareto fronts within QUBO- and Ising-based optimization frameworks. In this way, multi-objective problem formulations can be reconciled with the inherently single-objective nature of QUBO models in a principled and consistent manner using QUBO and Ising machines.

9.3.8 Augmented Lagrangian Treatment of Constraints

Most combinatorial optimization problems of practical relevance are subject to hard constraints, such as capacity limits, trust thresholds, or feasibility requirements. When such problems are reformulated in QUBO or Ising form, constraints cannot be enforced explicitly and must instead be incorporated into the objective function through penalty terms. While this transformation enables the use of quantum annealers and quantum-inspired optimization algorithms, it introduces a critical challenge: the calibration of penalty parameters.

Constraints and penalty-based QUBO formulations. Consider a generic constrained binary optimization problem of the form

$$\min_{x} \; f(x) \quad \text{subject to} \quad c_i(x) = 0, \; i \in \mathcal{E}, \qquad g_j(x) \leq 0, \; j \in \mathcal{I}, \tag{9.33}$$

where x denotes binary or spin variables, $f(x)$ is the objective function, and $c_i(x)$ and $g_j(x)$ represent equality and inequality constraints, respectively.

A standard approach to obtain an unconstrained QUBO formulation is to introduce quadratic penalty terms,

$$F(x) = f(x) + \sum_{i \in \mathcal{E}} \mu_i \, c_i(x)^2 + \sum_{j \in \mathcal{I}} \nu_j \, \max\big(0, g_j(x)\big)^2, \tag{9.34}$$

where μ_i and ν_j are positive penalty coefficients. In this formulation, feasible solutions correspond to low-energy configurations, while constraint violations are penalized energetically.

However, the effectiveness of this approach depends critically on the choice of the penalty parameters. If penalties are chosen too small, infeasible solutions may dominate the low-energy spectrum. If chosen too large, the resulting energy landscape becomes ill-conditioned, leading to numerical instability and degraded performance of QUBO and Ising solvers. As a consequence, penalty-based formulations often require extensive instance-dependent tuning, significantly limiting their scalability and robustness, particularly in the context of quantum and quantum-inspired optimization.

Augmented Lagrangian principle. The Augmented Lagrangian (AL) method provides a principled alternative for handling constraints by combining quadratic penalties with Lagrange multipliers [15]. For equality constraints, the augmented Lagrangian takes the form

$$\mathcal{L}_{\mathrm{AL}}(x, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i \, c_i(x) + \frac{\rho}{2} \sum_{i \in \mathcal{E}} c_i(x)^2, \tag{9.35}$$

where λ_i are Lagrange multipliers and $\rho > 0$ is a penalty parameter. Inequality constraints can be treated analogously through suitable transformations.

Unlike pure penalty methods, constraint satisfaction in the augmented Lagrangian framework is not enforced solely through large penalty coefficients. Instead, feasibility emerges through the joint evolution of the optimization variables, the multipliers, and the penalty parameters. This mechanism allows constraints to be enforced progressively, avoiding excessive distortion of the objective landscape and improving numerical conditioning.

Augmented Lagrangian methods in QUBO and Ising formulations. Recent work [49, 9], has shown that augmented Lagrangian techniques can be effectively combined with QUBO and Ising formulations by embedding the augmented Lagrangian objective into a sequence of unconstrained QUBO problems. Each QUBO instance is solved approximately, while Lagrange multipliers and penalty parameters are updated iteratively between solver calls.

In particular, the Augmented Lagrangian approach enables analytical or heuristic estimation of penalty multipliers, substantially reducing the need for empirical, instance-specific tuning. This strategy has been successfully demonstrated for constrained combinatorial optimization problems mapped to Ising machines, yielding improved robustness and feasibility compared to traditional penalty-only formulations :contentReference[oaicite:1]index=1.

Relevance for CASTOR optimization. Within the CASTOR framework, constraints arise naturally from trust requirements, policy restrictions, and network performance bounds. Encoding such constraints using fixed penalty coefficients would require careful and potentially brittle calibration, especially when combined with quantum-inspired solvers such as Simulated Bifurcation.

The augmented Lagrangian approach provides a scalable and principled mechanism for constraint handling, enabling smooth enforcement of feasibility while preserving the structure of the underlying optimization landscape. This makes it particularly well suited for hybrid classical–quantum and quantum-inspired optimization pipelines, where robustness, stability, and reduced parameter sensitivity are essential.

Chapter 10

User Stories for the overarching Trust Assessment Engineering process

10.1 Risk Engineering Process

Engineering Story-I

As the Service Orchestrator, I want to be notified of updated RTL for candidate path profiles based on the latest risk assessments and risk indices identified, so that I can select appropriately secure paths that account for cascading attacks risks and network topology criticality when deploying services.

Objective To enable the Facility Layer to make informed path selection decisions by incorporating dynamic risk analysis that reflects current risk assessments, including topology-aware metrics and external risk indices from neighbouring domains of frameworks where direct asset topology visibility may be unavailable.

Motivation Path security in CASTOR depends not only on the security of individual nodes, but also on their position within the network topology. Even a seemingly secure path may be risky if it traverses nodes that are critical to network operations. Moreover, risk conditions evolve dynamically, as new vulnerabilities are discovered, threat landscapes change, and in inter-domain scenarios, risk indices emerge from neighbouring domains or services where direct access to their asset topologies is unavailable. Without dynamic risk assessment that incorporates such external risk indices, path selections may rely on outdated or incomplete risk information. When direct observation of external domain topologies is limited, Monte Carlo simulation methods enable probabilistic risk evaluation by sampling from available risk evidence distributions, providing statistically robust RTL estimations even under uncertainty. CASTOR addresses this through continuous risk assessment that combines topological criticality metrics with the latest risk indices from both internal and external sources, leveraging Monte Carlo methods for uncertainty quantification in black-box scenarios, enabling RTL updates that reflect the risk landscape.

Requirements The Facility Layer assumes that it will be notified when updated RTL values become available for path profiles based on the latest risk assessments. Risk assessments must incorporate topology-aware metrics, including critical node scores based on network position, with individual node vulnerability assessments correlated with the topological context to identify paths that minimize exposure. The risk assessment methodology must support the dynamic integration of risk indices as they are identified, including external risk indices from neighbouring domains or infrastructure services (as specified in Use Case 4) where complete asset topology information is unavailable. RTL values must be expressed as subjective logic triplets compatible with the trust policy framework, enabling direct comparison with runtime ATL measurements during path enforcement. When incorporating external risk indices

with limited topology visibility, the risk assessment must provide appropriate uncertainty quantification while still reflecting available risk evidence in RTL calculations.

Engineering Story-II

As the Facility Layer, I want to receive updated RTL values that account for cascading attack effects when topology changes, so that path selections remain accurate and reflect the evolving risk of multi-hop attack propagation across interconnected network elements.

Objective To ensure the Facility Layer maintains accurate RTL information as network topology evolves by incorporating cascading attack analysis that models how the compromise of one node can propagate through network dependencies, enabling path profile assessments to account for temporal attack spread beyond isolated node vulnerabilities.

Motivation RTL accuracy in CASTOR depends on understanding not just individual node risks but also how attacks can cascade through the network topology. When network topology changes, for instance nodes join or leave, links fail, or connectivity patterns shift, the potential for cascading attacks evolves accordingly. A previously low-risk node may become a critical cascade point if topology changes elevate it to a key junction position. Similarly, the compromise of a single node can trigger cascading effects where the attack propagates to neighbouring nodes through network dependencies, creating risks that extend far beyond the initial compromise point. Without dynamic analysis of cascading effects that responds to topology changes, RTL values become stale and inaccurate, failing to reflect how topology evolution alters attack propagation paths. CASTOR addresses this through Markov chain-based cascading attack analysis that models multi-hop attack propagation probabilities and temporal spread dynamics. This enables RTL derivations to account for how network topology influences cascading failure scenarios, ensuring trust level calculations remain accurate as the network evolves.

Requirements The Facility Layer assumes it will receive updated RTL values triggered by significant network topology changes. Cascading attack analysis must employ Markov chain modeling to represent state transitions as attacks propagate from compromised nodes to adjacent nodes over time, with transition probabilities reflecting exploitation likelihoods based on vulnerability profiles and network dependencies. The analysis must support fault tree integration to identify critical cascade paths where single node compromises can trigger widespread failures. RTL updates must occur dynamically in response to topology changes, with recalculation triggered when nodes join/leave the network or connectivity patterns change significantly. The cascading analysis must model multi-hop propagation scenarios using absorbing Markov chains to determine steady-state compromise probabilities across network paths. Updated RTL values must maintain expression as subjective logic triplets, with uncertainty components reflecting the probabilistic nature of cascading attack predictions. RTL recalculation must complete within timeframes suitable for operational decision-making to ensure the Facility Layer can respond to topology changes without service disruption.

10.2 Trust Engineering Process

Engineering Story-III

As the Service Orchestrator, I want to bootstrap trust in the network topology, so that I can establish control plane interactions with elements that can demonstrate their trustworthiness.

Objective This ensures that a new network element meets the minimum trust requirements of an administrative domain, enabling it to access and download trust configuration and provision its in-device trust enablers (i.e., the TNDE). This, in turn, allows the Service Orchestrator to characterize the trust

posture of the network element and its links, both when deploying new services and when evaluating the trustworthiness of the network topology for forwarding service workloads (see Engineering Story-I).

Motivation

In CASTOR, Trust is conceived as a function that maps potential (dis)trust between different actors into a systematically modelled trust relationship. On the one hand, CASTOR considers trust functions among different entities both in the management plane (i.e., Service Orchestrator trusts network elements and their derived links) and in the forwarding plane (i.e., Network element A trusts network element B). This clearly states the subjectivity of trust as different actors may have different trust characterization over a common trust proposition. On the other hand, in addition to being subjective, trust is also context-dependent (see Section 5.4). Specifically, trust characterization for a network element may vary due to various reasons, including (i) the property under evaluation (e.g, integrity, availability), (ii) the trust relationships that affect the trustworthiness of a given entity, and (iii) the required trust level that needs to be attained at a given phase of the operational lifecycle of the topology. Consequently, this introduces the need to consider trust functions that span across the lifecycle of the network topology.

Before a network element is allowed to access any Trust Policy, it shall demonstrate to the Service Orchestrator its ability (i.e., it has the necessary mechanisms) to securely monitor, process, and report trustworthiness evidence related to its configuration and operational state. This constitutes an integral trust function that allows a network element to attest to the correctness of its Trusted Computing Base (TCB), which is a mandatory prerequisite defined in the secure onboarding protocol, described in D3.1 [7].

Upon successful completion of the attestation of the TCB platform, the network element is provisioned in order to start sharing its trustworthiness claims with its neighbourhood (i.e., in the forwarding plane). This second trust function allows for the (mutual) bootstrapping of trust within the network topology, allowing network elements to establish communication links with each other if and only if they meet the minimum trust requirements that are specified by the domain administrator. In CASTOR, this process is fully aligned with the IETF's Trusted Path Routing paradigm [4] and Engineering Story-IV discusses how it can be extended to incorporate runtime trustworthiness evidence, enabling the maintenance of the established trusted topology throughout its operational lifecycle.

In parallel to the exchange of evidence in the forwarding plane, the successful completion of the secure on-boarding process allows the Service Orchestrator to establish secure control-plane communication channels with the newly onboarded network element and to provision the cryptographic material required for the element's interaction with the rest of the CASTOR framework. The network element can then interact with the CASTOR Blockchain (see D5.1 [6]) to retrieve the Trust Policy corresponding to its assessed security posture. The enforcement of the Trust Policy will dictate the entire in-device trust engineering process: from the collection of the critical traces that are associated with the target router function under evaluation, to the configuration of the Trust Sources so that they process the traces and share trustworthiness evidence either with the Local TAF agent or the Orchestrator's Global TAF. Through the Trust Policies, CASTOR envisions to define a robust mechanism for specifying multiple Trust Functions both at the management and at the forwarding plane that are able to capture the behaviour of the entire network topology throughout its lifespan.

Requirements The selection of the appropriate Trust Policy for a network element depends heavily on its security posture. CASTOR performs continuous and comprehensive risk analysis to determine the details of each Trust Policy, including which trustworthiness evidence must be collected at runtime and the required trust level (RTL) associated with a specific target trust proposition (see Engineering Story-I). As explained in Engineering Story-II, the RTL for a given Trust Policy may also be adjusted based on the network element's position in the topology. This adjustment helps mitigate cascading attacks that could increase risk, ensuring that stronger guarantees are enforced during runtime.

Engineering Story-IV

As the Service Orchestrator, I want to be able to continuously evaluate the trustworthiness of the network topology and detect any changes in its state, so that I can maintain a trusted topology throughout its operational lifecycle.

Objective To extend trust evaluations beyond enrolment time in order to get runtime guarantees on the operational state of the entire network topology. This is essential because it allows the Optimization Engine to derive accurate recommendations based on the runtime trust characteristics of each element in the topology. It is also required for evaluating the trustworthiness of already-provisioned traffic-engineering policies as part of the service-assurance process.

Motivation Existing work towards trust-aware traffic engineering focuses on the establishment of "trusted topologies". On the one hand, the IETF's Trusted Path Routing paradigm [4] delineates the core system model, allowing only attested and trustworthy network devices to participate in routing decisions. In this context, each network element is evaluated prior to its inclusion in a trusted network domain. However, even though it highlights the need to "maintain" trust throughout the operational lifecycle of the network topology, the current specification does not delve into the architectural designs and challenges of such runtime trust monitoring capabilities. On the other hand, the SCION framework [12] follows a different approach by introducing a completely revised inter-domain routing protocol, designed to provide route control, failure isolation, and trust information for end-to-end service provisioning. Even though it specifies the concrete mechanisms to establish and update trust "agreements" dynamically (in the form of Trust Root Configurations - TRCs - within a specific set of autonomous systems called the Isolation Domain), it does not provide the means to systematically measure and evaluate trustworthiness. To this end, CASTOR seeks to bridge this gap by focusing on the key challenges required for end-to-end trust characterization at the node, path, and domain levels. Drawing from our initial analysis [30, 20], we highlight in the following the core challenges we have identified so far and the ways in which the CASTOR Trust Assessment Framework aims to address them:

- **Dynamic ATL Expression:** As highlighted in [20], one core challenge towards accurate trust evaluations lies in the correct modelling of the trust proposition (e.g., the router's configuration integrity has not been compromised) that we need to measure. This relates to the types of (runtime) evidence that need to be collected but also to the trust relationships that need to be considered. In both dimensions, it is crucial that a TAF instance is able to cope with dynamic updates stemming from the topology. For example, fresh attestation evidence may show a corruption in the network configuration of a router, whereas the addition of new router elements in the topology would require the inclusion of new trust relationships in order to get the overall trust level for the entire topology. Therefore, CASTOR envisions a robust TAF architecture that *(i) facilitates the collection of fresh runtime evidence (through the TSM), (ii) enables updates to the relevant Trust Models, and (iii) generalizes the TLEE capabilities to dynamically derive the ATL expression based on the latest trust model instance.*
- **Robust Modelling of Uncertainty:** Being able to reason about the trustworthiness of the topology is one of the core requirements that led CASTOR to adopt the Subjective Logic paradigm (see Section 5.3). Uncertainty hinges on multiple factors, including the relevance of trust sources, the suitability of the selected method, and the nature of the input data used by these methods [20]. In this context, *CASTOR will investigate different trust quantification methods as part of the TSM in order to evaluate the impact of different pieces of evidence on the uncertainty and the overall trust opinion to be derived*. For example, depending on how relevant a type of evidence is, the absence of corresponding observations may either increase the overall uncertainty or reduce the perceived belief in the associated trust opinion.

- **Consistent and Accurate RTL and ATL Values:** An important aspect in reasoning about the trustworthiness of a trust object (e.g., a network element, end-to-end path, or domain) is the comparison between its computed ATL and the predefined RTL constraints. In CASTOR, we adopt a risk-analysis approach as the basis to derive both ATL and RTL values (see Chapter 8), thereby providing shared semantics and enabling a meaningful comparison. As with ATL constraints, RTL constraints may also need to be updated at runtime. As mentioned in Chapter 7 this takes place as part of a Trust Policy update which could be the result of a revised risk analysis that led to the revision of the final RTL values. To this end, *CASTOR aims to design and implement the TDE subcomponent so that it can use the latest enforced Trust Policy, enabling trust decisions based on the most up-to-date and accurate ATL values.*

- **Convolution of Trust:** Monitoring the operational state of the underlying network topology presents challenges due to the variety and nature of runtime evidence that must be processed. Network elements can provide multiple sources of evidence, such as integrity monitors, behavioural analyses, or configuration compliance checkers. At the Global TAF level, this diversity necessitates careful modelling of all derived trust relationships and appropriate aggregation of the resulting trust opinions. To address this, *CASTOR envisions an expressive trust modelling approach, enabling dynamic management of trust relationships within the TSM subcomponent (see Section 7.1) and allowing the TLEE to apply the appropriate Subjective Logic operators—such as the fusion operator when consensus among multiple trust opinions over the same trust proposition is required.* Together, these mechanisms ensure reliable and efficient trust convolution across the network.

- **Evolution of Trust:** In order to enable runtime trust evaluations, it is essential to consider how new and existing knowledge can be incorporated into the assessment. CASTOR addresses this evolution of trust in a twofold manner. First, the notification model employed (e.g., asynchronous versus periodic updates) directly influences the freshness of evidence and, consequently, the derived trust opinion. Second, historical trust evaluations are integrated to account for past behaviour, providing resilience against temporary fluctuations and ensuring that previously penalized network elements require sustained positive evidence before regaining high trust. Together, these mechanisms enable the TAF to produce stable and context-aware trust assessments that reflect both recent observations and longer-term operational behaviour. To support both capabilities, *CASTOR's TAF architecture is envisioned to include a robust TSM subcomponent, capable of managing the temporal evolution of quantified trust opinions.*

Requirements As described in Section 7.1, all operations of the Trust Assessment Framework are governed by the enforced Trust Policy. This policy needs to be downloaded to any TAF instance (Global TAF or Local TAF agent). It primarily defines the Trust Model Template, specifying the trust relationships to be monitored and the sources of evidence that the TAF's TSM interacts with to dynamically collect runtime information on the configuration and behaviour of network elements. It also provides the details required by the TLEE, such as the types of SL operators (e.g., discounting or fusion for atomic trust propositions, or other logical expressions for composite ones, see Chapter 6), to compute the final ATL values. Finally, the Trust Policy may include RTL constraints, enabling the TDE to derive trust decisions with respect to the target trust propositions.

10.3 Optimization Engine Engineering Process

Engineering Story-V

As the Optimization Engine, I want to have access to the latest network- and trust-attributes that characterize each node/link in the topology graph, so that I can provide meaningful and accurate recommendations on path and alternative (explicit) paths per path profile.

Objective To ensure that the Optimization Engine operates on the latest topological graph enriched with current network- and trust-related attributes to enable accurate and trustworthy path recommendations per path profile for the Service Orchestrator.

Motivation The quality of the optimization results provided by the Optimization Engine to the Service Orchestrator directly depends on the freshness and consistency of the underlying topological information. Both the network- and trust-related metrics for the network evolve continuously in CASTOR based on incessant runtime measurements. CASTOR achieves this objective by integrating the Optimization Engine with mechanisms that continuously provide it with updated attributes, which are then consumed by the optimization engine at high frequency.

The Global Trust Assessment Framework (TAF) may provide either fresh evidence or cached evidence when populating trust-related attributes. However, the use of cached (and potentially obsolete) evidence can result in a topology view that does not fully reflect the most recent network or trust state. Nevertheless, obsolete data do not necessarily imply a violation of the Service-Level Agreement (SSLA). The cache data may reduce the optimality or confidence of the resulting path recommendations while still remaining within acceptable SSLA bounds.

Requirements The Optimization Engine assumes access to a logically consistent topology graph with attributes related meta-data. Both network and trust must be provided in a quantitative form suitable for multi-objective optimization, as per the problem formulation. The Optimization Engine depends on other CASTOR components to provide up-to-date node- and link-level trust attributes, and along with network-related attributes capturing the network state. Optimization Engine must be able to accommodate incremental updates with full topology reconstruction for high resource efficiency.

Engineering Story-VI

As a Service Orchestrator, I want to have access to the near-optimal set of solutions that can realize each path profile in my service catalogue, so that I ensure the selection of paths (active and backup) that accommodate all network- and trust-related attributes in an efficient manner.

Objective To enable the Service Orchestrator to obtain a bounded set of near-optimal paths (active and backup) that jointly satisfy the network performance objectiveness and trustworthiness for each path profile.

Motivation The trusted path routing problem within CASTOR is a multi-objective optimization problem that must jointly optimize over the network- and trust-related metrics. A single “best” active path is often insufficient to ensure service continuity. Therefore, providing performance and trust realization require alternate backup paths that are available without the need for the re-optimization. CASTOR’s attains this objective by leveraging Optimization Engine to compute not one but a set of near-optimal paths per path profile using a pre-defined methodologies; using either an exact or heuristic algorithm, as described in Section 9.3. Therefore, instead of recomputing a new path every time the active path becomes unavailable or unsuitable, the Service Orchestrator can deploy the alternate path that still meets the path profile requirement immediately.

We will address the following questions concerning the Optimization Engine in the upcoming deliverables.

Most importantly, based on empirical analysis, CASTOR investigates how and when the Optimization Engine should be executed within the framework. We decompose the design space of the Optimization Engine into a set of key, interrelated challenges that will shape its functional specification in D4.2 and beyond as follows.

- **Periodic vs. Asynchronous**: The first fundamental question concerns whether the Optimization Engine should operate periodically (near-continuously) or should be triggered asynchronously. The asynchronous trigger could, for example, come from explicit requests from the Traffic Engineering Policy Engine. The two approaches present different benefits and limitations in terms of responsiveness, stability, and computational cost. CASTOR will explore these trade-offs experimentally. The goal of these experiments would be to identify execution models that best balance agility with operational efficiency.
- **Time as an Optimization Constraint**: CASTOR treats the execution time of the Optimization as a critical first-class constraint. Therefore, the optimization algorithms should preferably support early termination or interruption. The early termination enables the return of partial results when strict timing constraints are reached. CASTOR will explore how such time-bounded execution and intermediate solutions affect both the accuracy of the optimization outcome and the efficiency of the resulting enforced policies.
- **Optimization Efficiency (Computational Reuse)**: Computing the entire network state from scratch for every local change could be computationally inefficient. Therefore, CASTOR will investigate mechanisms for reusing previous computations and performing incremental or localized (re) optimization. The computational reuse will minimize computational overhead while maximizing runtime performance and scalability.
- **Time of Result vs. Time of Enforcement**: CASTOR makes a clear distinction between the time at which candidate paths are computed and the time at which they are enforced in the data plane. This distinction introduces the need for robust post-processing capabilities to classify computed paths into active, backup, and restoration paths. This challenge is timely in light of ongoing IETF work on Circuit-Style Segment Routing (SR) Policies [41]. The work distinguishes the path into two types: Protection and Restoration. A Protection path is fully established in the data plane and ready to carry traffic. A Restoration path may be computed and partially established, but is not immediately ready for use. It is non-trivial to ensure that the enforced paths in a highly dynamic environment remain timely and valid. CASTOR aims to address this challenge by enabling an end-to-end pipeline. As further elaborated in Deliverable D5.1 [6], this pipeline originates at the Traffic Engineering Policy Engine, which leverages the output of the Optimization Engine to derive the routing policies to be enforced on the routers. Deliverable D5.1 [6] analyzes the possible strategies for mapping optimization results into forwarding-plane behaviour, including the enforcement of traffic engineering policies (e.g., explicit paths or dynamic configuration as part of the Segment Routing paradigm) via a Network Controller, as well as the explicit establishment of concrete paths through existing control-plane routing protocols (e.g., the Path Computation Element Protocol).

Requirements The Service Orchestrator assumes the existence of an Optimization Engine capable of producing multiple near-optimal solutions per path profile. Therefore, the Optimization Engine must expose the candidate paths per path profile in a form consumable by the Service Orchestrator. On the other hand, the path profiles defined in the service catalogue, submitted as an input to the Optimization Engine, must be interpretable as optimization objectives and constraints, as defined by the problem formulation. The Optimization Engine to provide valid solution must have access to the accurate and up-to-date network and trust attributes. The Optimization Engine must bound the size of the returned solution set as per the resource and time constraint to ensure scalability and operation feasibility. The Optimization Engine assumes that other CASTOR components are capable of enforcing it results consistently across the network without fail.

Chapter 11

Summary and Conclusions

This deliverable has established the conceptual and methodological foundations of the CASTOR Trust Assessment Framework (TAF), providing a structured understanding of trust and trustworthiness and highlighting the limitations of traditional trust verification approaches. Key terminology and core components were introduced, along with a detailed examination of the various trust assessment modalities envisioned in CASTOR, distinguishing between local, in-router evaluations and global, orchestration-level assessments. The report has explored the state-of-the-art in trust management, focusing on probabilistic and logic-based frameworks capable of handling uncertainty and conflicting evidence, ultimately motivating the adoption of Subjective Logic as the primary reasoning framework. It has further identified the trust relationships that underpin trust-aware traffic engineering policies, elaborated on the concepts of Actual and Required Trustworthiness Levels, and analyzed the necessity of a continuous risk-aware evaluation process. By framing trust-aware traffic engineering as a multi-objective optimization problem, the deliverable has synthesized network and trust attributes into a coherent modeling approach. Finally, the survey of exact and heuristic optimization methodologies has provided the analytical grounding for the design and implementation of the CASTOR optimization engine.

Building upon this foundation, the deliverable has realized a concrete set of engineering stories that capture the main challenges, requirements, and design considerations for both risk- and trust-aware network operations. These stories serve as actionable guidance for the functional specification of the CASTOR TAF and its optimization engine, bridging the gap between conceptual models and practical system implementation. They form the basis for the next stage of development, providing a clear roadmap for translating trust and risk analysis into deployable traffic engineering policies. Consequently, the insights and specifications presented here will directly inform the design, implementation, and validation work in the subsequent deliverable, ensuring a smooth and structured progression from conceptual foundations to operational realization.

Bibliography

[1] A. Abbas et al. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, pages 1–18, 2024.

[2] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, 21(2):1851–1877, 2019.

[3] Khalil Amine. Multiobjective simulated annealing: Principles and algorithm variants. *Advances in Operations Research*, 2019(1):8134674, 2019.

[4] H. Birkholz, E. Voit, C. Liu, D. Lopez, and M. Chen. Trusted Path Routing. https://www.ietf.org/archive/id/draft-voit-rats-trustworthy-path-routing-11.html, January 2025.

[5] CASTOR. Operational landscape, requirements and reference architecture - initial version. Deliverable 2.1, The CASTOR Consortium, 14 2025.

[6] CASTOR. Architectural specification of dynamic enforcement of trust-/network-aware path establishments. Deliverable 5.1, The CASTOR Consortium, 15 2026.

[7] CASTOR. Conceptual architecture of castor trusted computing base & composable attestation model specification. Deliverable 3.1, The CASTOR Consortium, 15 2026.

[8] CASTOR. Trusted path establishment building blocks, optimization engine & crypto structures for trusted data sharing. Deliverable 4.2, The CASTOR Consortium, 18 2026.

[9] Lorenzo Cellini, Antonio Macaluso, and Michele Lombardi. Qal-bp: an augmented lagrangian quantum approach for bin packing. *Scientific Reports*, 14(1):5142, 2024.

[10] Federico Cerutti, Lance M. Kaplan, Timothy J. Norman, Nir Oren, and Alice Toniolo. Subjective logic operators in trust assessment: an empirical study. *Information Systems Frontiers*, 17(4):743–762, August 2015.

[11] Shao-Hen Chiew, Kilian Poirier, Rajesh Mishra, Ulrike Bornheimer, Ewan Munro, Si Han Foon, Christopher Wanru Chen, Wei Sheng Lim, and Chee Wei Nga. Multiobjective optimization and network routing with near-term quantum computers. *IEEE Transactions on Quantum Engineering*, 5:1–19, 2024.

[12] Laurent Chuat, Markus Legner, David Basin, David Hausheer, Samuel Hitz, Peter Müller, and Adrian Perrig. *The Complete Guide to SCION: From Design Principles to Formal Verification*. Information Security and Cryptography. Springer Cham, 1 edition, 2022.

[13] Pedro Maristany de las Casas, Antonio Sedeno-Noda, and Ralf Borndörfer. An improved multiobjective shortest path algorithm. *Computers & Operations Research*, 135:105424, 2021.

[14] Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232, 1968.

[15] Kangkang Deng, Rui Wang, Zhenyuan Zhu, Junyu Zhang, and Zaiwen Wen. The augmented lagrangian methods: Overview and recent advances. *arXiv preprint arXiv:2510.16827*, 2025.

[16] Jean Dezert and Albena Tchamova. On the validity of dempster's fusion rule and its interpretation as a generalization of bayesian fusion rule. *International Journal of Intelligent Systems*, 29(3):223–252, 2014.

[17] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pages 287–290. 2022.

[18] European Commission, DG Communications Networks Content and Technology. White paper – how to master europe's digital infrastructure needs? White Paper COM(2024) 81 final, European Commission, February 2024. Accessed: 2026-1-26.

[19] European Union Agency for Cybersecurity. Interoperable EU Risk Management Toolbox. Technical report, ENISA, 2023. Accessed: January 23, 2026.

[20] Nikolaos Fotos, Koffi Ismael Ouattara, Dimitrios S Karas, Ioannis Krontiris, Weizhi Meng, and Thanassis Giannetsos. Actions speak louder than words: Evidence-based trust level evaluation in multi-agent systems. In *International Conference on Information and Communications Security*, pages 255–273. Springer, 2025.

[21] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[22] Fred Glover, Gary Kochenberger, Rick Hennig, and Yu Du. Quantum bridge analytics i: a tutorial on formulating and using qubo models. *Annals of Operations Research*, 314(1):141–183, 2022.

[23] Hayato Goto, Kotaro Endo, Masaru Suzuki, Yoshisato Sakai, Taro Kanao, Yohei Hamakawa, Ryo Hidaka, Masaya Yamasaki, and Kosuke Tatsumura. High-performance combinatorial optimization based on classical mechanics. *Science Advances*, 7(6):eabe7953, 2021.

[24] Hayato Goto, Kosuke Tatsumura, and Alexander R Dixon. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems. *Science advances*, 5(4):eaav2372, 2019.

[25] Hiroshi Ikeda and Takashi Yamazaki. Multi-objective optimization technique based on qubo and an ising machine. *IEEE Access*, 12:8957–8969, 2024.

[26] Audun Jøsang. *Subjective logic*, volume 3. Springer.

[27] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(03):279–311, 2001.

[28] Audun Jøsang and Simon Pope. Dempster's rule as seen by little colored balls. *Computational Intelligence*, 28(4):453–474, 2012.

[29] Hongzhaoning Kang, Gang Liu, Quan Wang, Lei Meng, and Jing Liu. Theory and Application of Zero Trust Security: A Brief Survey. *Entropy*, 25(12):1595, November 2023.

[30] Ioannis Krontiris, Thanassis Giannetsos, and Henk Birkholz. Runtime trusted platform reporting (tpr). Internet Draft draft-rats-runtime-tpr-00, IETF Network Working Group, July 2025. Work in Progress, Intended Status: Informational; expires 8 January 2026.

[31] Christof Külske. The ising model: highlights and perspectives: C. külske. *Mathematical Physics, Analysis and Geometry*, 28(3):20, 2025.

[32] Barry M McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Harvard University Press, 1973.

[33] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379, 2022.

[34] National Institute of Standards and Technology (NIST). NIST Special Publication 800-39: Managing Information Security Risk: Organization, Mission, and Information System View. Special Publication 800-39, NIST, 2011. Accessed: January 23, 2026.

[35] Nils J Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[36] Vilém Novák, Irina Perfilieva, and Jiri Mockor. *Mathematical principles of fuzzy logic*, volume 517. Springer Science & Business Media, 1999.

[37] E. Pelofske, A. Bärtschi, and S. Eidenbenz. Quantum annealing vs. qaoa: 127 qubit higher-order ising problems on nisq computers. In *High Performance Computing*, pages 240–258. Springer Nature Switzerland, 2023.

[38] Rodolfo A Quintero and Luis F Zuluaga. Qubo formulations of combinatorial optimization problems for quantum computing devices. In *Encyclopedia of Optimization*, pages 1–13. Springer, 2022.

[39] Finley Alexander Quinton, Per Arne Sevle Myhr, Mostafa Barani, Pedro Crespo del Granado, and Hongyu Zhang. Quantum annealing applications, challenges and limitations for optimisation problems compared to classical solvers. *Scientific Reports*, 15(1):12733, 2025.

[40] Hai-Jun Rong, Plamen P Angelov, Xiaowei Gu, and Jian-Ming Bai. Stability of evolving fuzzy systems based on data clouds. *IEEE Transactions on Fuzzy Systems*, 26(5):2774–2784, 2018.

[41] Christian Schmutzer, Zafar Ali, Praveen Maheshwari, Reza Rokui, and Andrew Stone. Circuit Style Segment Routing Policy. Internet-Draft draft-ietf-spring-cs-sr-policy-13, Internet Engineering Task Force, December 2025. Work in Progress.

[42] Antonio Sedeño-Noda and Marcos Colebrook. A biobjective dijkstra algorithm. *European Journal of Operational Research*, 276(1):106–118, 2019.

[43] Sven Seuken and David Parkes. Sybil-proof accounting mechanisms with transitive trust. 2014.

[44] Glenn Shafer. A mathematical theory of evidence. 2020.

[45] Shubhkirti Sharma and Vijay Kumar. A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering*, 29(7):5605–5633, 2022.

[46] Philippe Smets. Practical uses of belief functions. *arXiv preprint arXiv:1301.6741*, 2013.

[47] Lotfi A Zadeh and Anca Ralescu. On the combinability of evidence in the dempster-shafer theory. In *Proceedings of the Second Conference on Uncertainty in Artificial Intelligence*, pages 347–349, 1986.

[48] Qing-Guo Zeng, Xiao-Peng Cui, Bowen Liu, Yao Wang, Pavel Mosharev, and Man-Hong Yung. Performance of quantum annealing inspired algorithms for combinatorial optimization problems. *Communications Physics*, 7(1):249, 2024.

[49] Gan Zheng and Ioannis Krikidis. Constrained higher-order binary optimization for wireless communications systems using ising machines. *IEEE Transactions on Wireless Communications*, 2025.