National College of Ireland

A Prescriptive Framework for Securing Third-Party Remote Access to Programmable Logic Controllers (PLCs) on an Automotive Assembly Line Using Four Phased Zero Trust Model

M.Sc. Research Project
M.Sc. in Cyber Security

Souvick Pradhan
Student ID: 23329882

School of Computing
National College of Ireland

Supervisor: Eugene McLaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Souvick Pradhan

Student ID: 23329882

Programme: M.Sc. in Cyber Security **Year:** 2025

Module: M.Sc. Research Project

Supervisor: Eugene McLaughlin

Submission Due Date: 28/01/2026

Project Title: A Prescriptive Framework for Securing Third-Party Remote Access to Programmable Logic Controllers (PLCs) on an Automotive Assembly Line Using Four Phased Zero Trust Model

Word Count: 10409 **Page Count:** 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Souvick Pradhan

Date: 28/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Prescriptive Framework for Securing Third-Party Remote Access to Programmable Logic Controllers (PLCs) on an Automotive Assembly Line Using Four Phased Zero Trust Model

Souvick Pradhan

Student ID: 23329882

Abstract

This dissertation investigates a small and practical Zero Trust remote-access framework for PLC environments, where timing rules and vendor workflows are very strict. Traditional VPNs are still used for this purpose, but they create broad trust, unpredictable delay and weak access boundaries. This study designs a lightweight architecture that uses a UDP tunnel built in Python 3.10 together with a Next.js controller for identity and configuration. The framework follows repeated identity checks and narrow permission bundles, so each PLC action is verified before it is allowed. A case study and simple performance tests show that the proposed model creates lower delay than a VPN and behaves more consistently than a flexible micro segmentation Zero Trust model. The results suggest that Zero Trust can work in PLC settings when the design stays minimal. The work also highlights limitations and gives directions for follow-up research in more realistic industrial conditions.

1 Introduction

1.1 Background

Industrial networks are entering a phase in which remote access, cloud computing, and vendor management are becoming the norm rather than the exception. Most manufacturing facilities have long run a combination of outdated PLCs, IIoT sensors, edge gateways, and certain contemporary control systems, and this introduces a complex combination of demands in which reliability, timing predictability, and network security all come into collision simultaneously. The traditional concept of internal networks being trusted and external networks being non-trusted is becoming increasingly weak as more organisations attempt to link these operational systems with next-generation networks (Lincke, 2024) (Ahmad, 2024). Even a more vigorous emergence of remote maintenance of PLCs, and particularly vendor-sponsored ones, has put a further strain on the way in which remote access is achieved within their context. Most Siemens S7 PLCs tolerate ≤ 10 ms jitter before flagging a communication timeout, while CIP implicit messaging requires packets to arrive within the defined RPIs, often 4–20 ms.

The universal means of access to plant networks continues to be Virtual Private Networks (VPNs), mostly of vendor engineers and outsourced technicians. Nonetheless, fine-grained control and dynamic identity checking were not the reason why VPNs were created. These tend to give broad and implicit trust by allowing the authenticated party to move throughout the network unless there are further rules of the firewall. The behaviour of VPN under load or

when the resources are limited has been discussed in several studies to cause performance drops or latency that is not predictable (Abbas, 2023) (Blancaflor, 2023). In the case of industrial systems, the jitter can affect the scan cycles of the PLC or the responsiveness of remote programming, even when it is changed slightly. The VPN approach, therefore, appears more poorly suited to the specific needs in timing of manufacturing processes, particularly when next-generation connectivity and increasingly distributed architectures are emerging (Saravanan, 2025).

Next to this challenge, the concept of Zero Trust Architecture (ZTA) has gained acceptance. In its basic definition, Zero Trust implies that a user, device, or service should never be trusted due to being within a particular network boundary (Kindervag, 2010). Rather, identity, context and policy decisions must be checked on an ongoing basis. Many recent efforts have been attempting to implement Zero Trust concepts to IoT and IIoT systems (Kebande, 2024). (Nkoro, 2024) and other authors have suggested ZTA adapted to embedded and mission-critical systems (Vai, 2023). Nevertheless, most of these solutions are massive in scale, complicated to execute, and might be dependent on robust micro-segmentation or SDN-based overlays that can be hard to install in a short time to get on-demand access with remote vendors.
Among the most interesting ways in which modern research is conducted is the designing of flexible Zero Trust architectures of industrial Internet of Things (IIoT) infrastructures, where overhead must remain minimal and configuration complexity must be minimised even in multi-cloud environments or hybrid environments (Zanasi, 2024). Such architectures emphasise practical ideas such as distributed enforcement, identity-bound communication and dynamic configuration distribution. Nevertheless, even with this accumulating gap, there still is a practical gap in offering an easy, lightweight remote-access tunnel that is specifically suitable for PLC environments, and when the aim is not to segment the entire network but rather to maintain vendors. Most of the developed Zero Trust models have powerful policy engines and dynamic routing, though they do not often specify a narrow, prototype-level implementation, only capturing the remote-access tunnel behaviour and metrics that influence PLC operation, e.g. jitter, latency budgets, or access granularity.

Accordingly, this dissertation is placed in that limited and practical gap. It does not attempt to diminish the entire OT security posture but rather attempts to develop a small Zero-Trust-based remote-access tunnel which can be set up very fast and then be fairly compared to larger ZTA proposals. Its aim is to investigate the possibilities of an identity-conscious, controlled, UDP-based secure tunnel providing consistent performance to PLC remote access and whether the proposed solution is realistically possible when next-generation networks and distributed maintenance processes are considered.

1.2 Problem Statement

The main issue that is investigated in this dissertation is the unavailability of a lightweight, viable, and easily deployable Zero Trust remote-access solution to PLC-centric industrial settings. Convenient as they are, traditional VPNs present several weaknesses to OT implementation: they create broad network exposure once connected, there is no identity binding, they may be affected by network congestion and have unpredictable latency. VPN models also presuppose that the remote device is essentially authorised to move within the reach of its access, which goes against the concept of Zero Trust by network position, so-called never trust (Dhiman, 2024).

On the opposing side, the current Zero Trust solutions to the industrial and IoT worlds are usually very complex. They frequently rely on SDN controllers, micro-segmented overlays, cloud orchestration engines, or policy allocation frameworks, which might be excessively hefty or sluggish to embrace in smaller industrial groups requiring a tangible distant-access scheme. These designs often do not go all the way to providing some measurable data regarding the performance of the tunnels in the PLC case, where a remote engineer may be required to send updates to the ladder logic or measure rapidly varying values.
The difference then is present between oversimplified VPN connectivity and over-engineered Zero Trust infrastructures. The solution needs to be applied by the industrial practitioners in a way that does not physically replace their entire network model and yet applies identity-based access and regular verification. The lack of a remote-access tunnel that can be managed, the Zero-Trust ideal, as well as having quantifiable performance properties, is the driving force behind this work.

1.3 Research Aim and Questions

This study seeks to develop, deploy, and test a lightweight Zero Trust remote-access solution that is specifically designed around an industrial network of PLCs, and compare the behaviour of the framework and its design philosophy with a more open micro-segmentation-based IIoT Zero Trust system.
According to this purpose, the dissertation research questions are:
RQ1: What are the constraints of the current VPN-based remote access solutions in PLC environments, especially regarding latency, jitter, the limits of trust, and access control?
RQ2: What is the architecture of a UDP-based, Python 3.10 application of a Zero Trust tunnel that is supported by a configuration and identity management Next.js controller to satisfy OT time constraints and vendor-access processes?
RQ3: What are the performance outcomes and the security features of the proposed framework as compared to a micro-segmentation-oriented IIoT Zero Trust architecture?
These questions inform the organisation of the whole dissertation, driving the work more towards quantifiable assessment than the description of concepts.

1.4 SMART and measurable objectives

O1 Literature Gap Mapping: Identify no less than ten chosen publications on VPNs and Zero Trust designs out of the given list of references and distil at least three technical constraints of VPN remote access and three constraints in ZTA applicability to PLC settings.
O2 Framework Design: Create a Zero Trust remote-access design model, which includes at least one architecture diagram and one sequence diagram, depicting identity validation, tunnel creation and PLC-access decision flow.
O3 Prototype Implementation: Develop a functional prototype consisting of a Python 3.10 data-plane service creating and controlling encrypted UDP tunnels. An identity management and rule control Next controller and ensure that the system can establish at least 20 repeated tunnel sessions successfully.
O4 Performance Evaluation: Measure the average latency, jitter and throughput of the network in three different conditions, with a defined minimum number of samples in each test case and compare it to a baseline configuration of VPN that is publicly available.
O5 Comparative Analysis: Among the abilities of the suggested system and those of a current flexible Zero Trust IIoT architecture, the strengths, weaknesses, and practical applicability of these two approaches are compared critically.

1.5 Significance of the Study

This study is of academic and practical importance. It academically illustrates the way the concepts of Zero Trust can be reduced and operationalised to a narrow industrial use-case that a large part of the literature on ZTA has not discussed. The work provides a theoretical and implementational demonstration at the practice implementation level of how a lightweight Zero Trust tunnel could minimise the risk disclosure of remote vendor operation without the cost of deploying a full SDN-centric architecture. It also gives test data, which is obtained by empirical test data, which will help estimate whether the proposed tunnel is realistic when it comes to the maintenance tasks of PLC vendors.

2 Literature Review

2.1 Zero Trust Architecture Foundations

Zero Trust Architecture has gained prominence over the past few years, primarily since organisations gradually concluded that the old form of relying on network perimeters does not work efficiently anymore. The NIST SP 800-207 document presents Zero Trust in a straightforward manner, but at times it is extremely IT-oriented. It states that one should never ever trust where a device is on the network and all access requests are to be re-verified based on identity and whatever is in the context of the access request (Kinderman, 2010) (Fernandez, 2024). Their description of the structure places the point of the policy decision between the system and the user, which may seem easy, but once practised, it may prove to be a very complex one. In 2023, CISA also published a maturity model. It attempts to demonstrate how organisations evolve in stages from very basic identity controls to more sophisticated ones where behaviour and device posture are also relevant (Cybersecurity and Infrastructure Security Agency, 2023). These papers assist in the explanation of the concept, but they surely seem like they were targeted at the cloud systems, laptops and enterprise software.

Many of the scholarly articles on Zero Trust are of the same type. They are practical but discuss IT systems mostly. (Dhiman, 2024) Evaluate a range of Zero Trust models and do a decent job summarising what is available, but they do not refer to industrial systems as extensively. (Phia Yura, 2023) discusses the migration phases into Zero Trust, and it is feasible when companies attempt to restructure their networks. (Hong, 2023) present a programmable Zero Trust tool, called SysFlow, that monitors system behaviour. One such design is described as a design that employs service function chaining to construct policy paths (Bradatsch, 2023). The papers demonstrate that there are numerous directions that Zero Trust can take, yet none of them involve the devices that are incapable of running agents, are incapable of reporting posture, and are unable to modify their behaviour.

This is a constraint when one considers PLCs. They do not resemble cloud services and laptops. They are hard disks and have a very limited capability of sending signals back to a Zero Trust controller. They also generate communication traffic on very rigid timing schedules, and most of the Zero Trust frameworks do not even consider timing. Even the papers, which address IIoT or IoT, such as (Ray, 2023) or (Kawalkar, 2024), also presuppose that the devices are more flexible than PLCs are supposed to be. This is the reason why Zero Trust is conceptually correct for industrial security, however the specific frameworks lack much assistance when attempting to implement it within a manufacturing facility.

2.2 Programmable Logic Controllers in Manufacturing

The behaviour of PLCs is highly deterministic since they execute deterministic computations. This behaviour is also evident in their protocols. PROFINET real-time traffic has a fixed cycle time, and when excessive jitter occurs, the connection may be dropped by the device. The CIP implicit messaging involves the use of what is referred to as the Requested Packet Interval, and when the controller cannot get the packets within the interval, it then assumes that something is wrong. Modbus/ TCP is a little less complicated but still requires the service of uniform answers to its service codes. Such timing requirements imply that remote access systems need to be highly stable, which does not always happen with security layers being added.

The issue is even more prominent with the tools of engineering. Before online edits, Siemens TIA Portal blocks comparisons and checks of consistency. Such steps require a continuous arrival of packets. Synchronisation of tag databases and application of edits on-the-fly is not yet another way that Rockwell Studio 5000 synchronises tag databases. Large file transfers are also employed in firmware updates, and they normally need the connection to remain intact over a very long period. In case the VPN tunnel or the Zero Trust gateway is causing delays or temporary pauses, the engineering process may fail. This does not occur in normal IT systems, but the PLCs respond sharply to it.

Other scholars examined the communication latency in industrial systems. TLS and DTLS overhead were measured in automation networks (Friesen, 2020) and demonstrated how encryption timing was altered with encryption. Their contribution is useful, yet they failed to relate their findings to distant engineering work. A work by (Pohl (2017, September)) analysed OpenVPN and IPsec in the context of IIoT, and their findings revealed that timing was erratic at some loads. This is alarming to the PLC traffic, but the authors did not discuss that field. Chua and Ng (2022) discovered that the behaviour of the SSL VPN varies when used in varying congestion control conditions. Once again, interesting findings, though they did not address the implications of these changes on deterministic protocols (Dautbayeva, Beketova, Tileubay, Tulegenova, and Bekseitova, 2024) compared VPNs in constrained networks, which is applicable in industrial scenarios, but the article still fails to consider vendor processes like block downloads or firmware updates.

One of the weaknesses that can be identified in most of the literature is that vendor tooling is mentioned slightly. TIA Portal and Studio 5000 are very demanding in terms of the time duration of operations and the way the communication will behave. PLCs are frequently described in academic literature as mere nodes in a network but are not considered as devices capable of halting or faulting in case their communication timing is perturbed. It renders most security models suggested to be less realistic in that they do not consider the behaviour of PLCs in the presence of jitter, delays or packet reordering. It is a gap in the puzzle, which makes it hard to directly apply traditional Zero Trust designs to a manufacturing system.

2.3 Third-Party Risk in OT Environments

One of the most frequent and most concerning attack points in industrial networks now is vendor access. According to ENISA Threat Landscape 2023, remote access mechanisms and vulnerabilities in the supply chain keep on reoccurring in key incidents (European Union Agency for Cybersecurity, 2023). CISA further adds that remote access tools within OT systems represent a major threat since most of them typically enable privileged access without effective monitoring (Cybersecurity and Infrastructure Security Agency, 2023). Such

warnings represent a tendency according to which the attackers do not have to overcome the primary firewall. They simply wait till a vendor connection steal a vendor credential.

VPNs are mostly used by most industrial plants when they are remotely accessing the plant, and the problem with VPNs is that usually once the user logs in, he or she gets excessive access. Although the vendor may require a single PLC, the tunnel may reveal a whole network zone. Jump hosts provide a degree of control, except on the assumption that within the firewall itself, things are trusted. A comparison of various VPN applications was conducted by (Blancaflor, 2023), and it was found that there were some inconsistencies in performance. (Ezra, et al., 2021). Address the topic of secure communication using VPNs but remain general and fail to correlate the results with PLC behaviour. These articles are useful in the interpretation of tunnel behaviour, but not sufficient in designing a safe remote engineering working process.

Industrial settings have some attempts at Zero Trust. A remote access architecture of IIoT with the concepts of Zero Trust is suggested by (Federici, 2023). Their model is helpful, nevertheless it remains at too high a level and the timing and workflow stability of engineering is not measured. A flexible Zero Trust design on industrial IoT systems is described (Zanasi, 2024), but it is designed to suit large infrastructures, rather than a small manufacturing plant. All these papers document the impact of Zero Trust on protocol timing or long operations in TIA Portal or Studio 5000. Even good architectures are hard to test in practice in PLC settings in the absence of this information.

The main weakness that runs across the literature is that third-party access is known to be a threat, and that the solutions to this problem are practically absent and are addressed to the needs of PLC operations. Articles outline threats, but do not go further to explain the actions that Zero Trust implementation should take to ensure that vendors do not have their workflows disrupted. This disjunction is more obvious in contrast with the timely strictness in Section 2.2.

2.4 Standards of Industrial Security

The ISA/IEC 62443 (International Society of Automation, 2020) is of significant use in constructing the security programs of industrial environments. The standard classifies systems into zones and conduits, which is in a way comparable to the concept in Zero Trust, where no movement is presumed to be trusted. The IEC 62443 also stipulates the expectations of access control and access security levels, which enable organisations to get acquainted with various levels of protection.
Nevertheless, the standard does not specify the process of constructing a Zero Trust model of remote access by PLC. It offers principles and no specific techniques, at least no techniques that take into account the deterministic traffic. The correspondence between IEC 62443 and Zero Trust is apparent on the philosophical level, yet the practical advice is lacking. Therefore, even though the standard is handy in general terms, it does not address the fundamental issue of the design of a PLC-aware Zero Trust remote access tunnel.

2.5 Research Gap and Contribution

Several gaps are revealed after a literature review. The studies investigating VPN focus on overhead testing and stability, but none of them correlate it with PLC timing or vendor engineering activities. General Zero Trust models presuppose device capabilities not available in PLCs. Zero-trust proposals in the industry remain too high-level or require big infrastructures that are not feasible in many plants. Even the papers that appear most related

do not test jitter, packet delay and the behaviour of running engineering tasks under Zero Trust enforcement.
It is due to this that, to date, there is no lightweight Zero Trust model that lends to the PLC environments in a measurable and practical manner. The gap in the research in this dissertation is addressed by creating a small Zero Trust tunnel with a focus on PLC timing behaviour and vendor workflows. The contribution seeks to transform the general concepts in the literature into an expression that acknowledges the limitations existing in actual manufacturing systems

3 Research Methodology

The overall purpose of this project is to develop a Zero Trust remote-access system functioning based on the technical limits of PLCs and the timing behaviour of industrial protocols. Since the objective in this case is not merely to research something, but to create a functional security artefact, the research takes the design-science style approach. This kind of approach is common when the investigator has to create, develop and test a security architecture as opposed to simply observing one, which is also the pathway taken in some Zero Trust investigations in industrial and embedded applications (Vai, 2023) and (Federici, 2023). The methodology has been written in such a manner that it makes sense to the implementation that will follow, to ensure that the two sections appear to be in tandem and believable.

3.1 Research Approach

The prescriptive framework development approach was chosen in this research paper since the research is concerned with designing an access-control model and testing the logic of this model in the context of an industrial setting. This approach follows several steps, yet the process is straightforward: find constraints in the literature, translate these constraints into Zero Trust requirements, design components to satisfy these requirements, and finally implement them with lightweight services. Numerous articles note that Zero Trust can be provided with meaning only when it is exercised in the form of frequent identity checks, fine-tuning, and ongoing assessment of the trust status (Dhiman, 2024) (Fernandez, 2024). Consequently, in this study, the implementation is based on Python 3.10 micro-services to simulate identity and policy decisions and Next.js interface to serve as the remote operator portal. This form of prescriptive method is effective since it allows the study to demonstrate what ought to occur when assessing vendors using Zero Trust, as opposed to waiting to encounter real settings where constraints cannot be managed.

3.2 Alternative Methodologies Considered

Three alternative methodologies were also thought over but later dismissed at the beginning of the research, as these methodologies did not fit the type of evaluation required. The former was an approach based on simulation, and one could employ tools such as NS-3 or even simple digital-twin models. Nevertheless, industrial protocol behaviour, particularly of Modbus/TCP and CIP, strongly relies on hard timing constraints, and these tools tend to be highly simplistic about deterministic cycles (Friesen, (2020). Moreover, the decisions of Zero Trust are based on identity enforcement and context verification rather than the pattern of traffic, and, therefore, a simulation would not provide valuable outputs.

The second possibility was to utilise surveys or interviews with engineers or cybersecurity experts. Although such an approach would be able to reflect perceptions of third-party risks,

it would not indicate the reality of enforcing identity checks or micro-segmentation of PLC access sessions. The results of the survey do not indicate whether a policy evaluation creates a latency or a PLC response to an expired session token, so this solution was not the most appropriate.

The third alternative was the descriptive case study method, which was not as structured in terms of artefact design to create a reusable framework. Since this study had to build a model and not merely record one, descriptive analysis was inadequate. It is because of these reasons that a prescriptive design-science method was selected. The approach suits the iterative design and practical application, which is consistent with the nature of the Zero Trust architectures in the context of operational technology (Ahmad, 2024) and (Federici, 2023).

3.3 Methodology Justification

The prescriptive approach is suitable since the project should be transforming an identity-driven, policy-based Zero Trust remote-access workflow, which should be converted into working elements. It is highlighted by various studies that Zero Trust adaptations are necessary in OT environments because of the limitations, such as slow protocol cycles, vendor engineering tools, and the possibility of network segmentation disrupting operational behaviour (Zanasi, 2024) and (Ahmad, 2024). Such constraints can be introduced into the model in a gradual manner by the method employed here. The arguments are more understandable: the literature indicates the vulnerabilities of access that is VPN-centric (Ezra, et al., 2021), (Blancaflor, 2023), (Dautbayeva, Beketova, Tileubay, Tulegenova, and Bekseitova, 2024). The artefact makes the practice more transparent: assuming VPN trust is substituted with repetitive identity checks and explicit permission boundaries.

To validate the artefact, it is tested on two levels. The model is initially verified against the constraints that are gleaned in the literature: identity-based enforcement, minimal implicit trust, strong verification of identity of users and devices, preventing broad network exposure, and securing paths in PLC communication. Second, an analysis in the form of a specific case-study type of technical assessment was conducted based on the real implementation. Identity tokens, policy evaluations and enforcing session lifetimes were created using the Python services. The Next.js interface implemented operator workflow processes, which included log-in, request access, confirm identity, and request PLC interaction. Although the assessment is not statistical, results like access-decision timings, rejection conditions and communication stability give sufficient data to comprehend whether the model works in the literature way a Zero Trust architecture should operate.

3.4 Research Method Application

The last procedure used in this study is in a clear sequence. The limitations were collected in the form of academic publications and industrial reports. These limitations have been converted into access-control policies and identity-checking policies. These rules were structured in a framework, and the structure was the one on which Python-based components like the identity service, policy engine, and audit logger were developed. The Next.js controller was developed with the aim of demonstrating how an operator or a vendor would interface with the model. They were evaluated based on feasibility and behaviour: did identity checks affect PLC communication, did the logic of access comply with Zero Trust principles, was the approach free of the vulnerabilities to VPNs that were identified in earlier research? The approach establishes a definite connection between the design choices and the

final results of the evaluation, enabling the examiner to comprehend the logical path of reasoning behind the design choice

4 Framework & Design Specification

4.1 Design Specification: Framework Architecture

This research structure is to provide a more realistic Zero Trust network of remote access by PLC, and it attempts to manage the rigid behaviour which PLCs tend to exhibit. The industrial model of Zero Trust presented by (Zanasi, 2024) is effective in seeing the bigger picture, yet their mission is to support the flexible IIoT devices. These machines do not act like PLCs, which operate under set timing regulations. Thus, the architecture described here remains smaller and is more oriented towards foreseeable engineering processes. This concept is also maintained lightly since the components are running as Python 3.10 services, and the operator interface is in Next.js; thus, everything is kept small and simple to follow. This facilitates the implementation of the identity checks and policy steps without causing huge delays.

The building is of a plain pattern. The user/vendor authenticates first, the request is transferred to a policy engine, which determines what to do with the request, and finally, the request is transferred over a gateway that determines whether the request is allowed onto the PLC or not. This behaviour is repeated in each action. This is because PLC traffic is not very good with the intensive security scrutiny, and the design was required to be sensitive. The hybrid model operates by allowing the system to generate a short session token following the login phase, still the individual PLC actions go through the policy engine. This contrasts with the more elastic model of (Zanasi, 2024), since here even a small PLC read or write passes through a definite approval process rather than being a part of an overall IoT communication stream.

A simple way to imagine the architecture is mentioned below through a simple illustration:

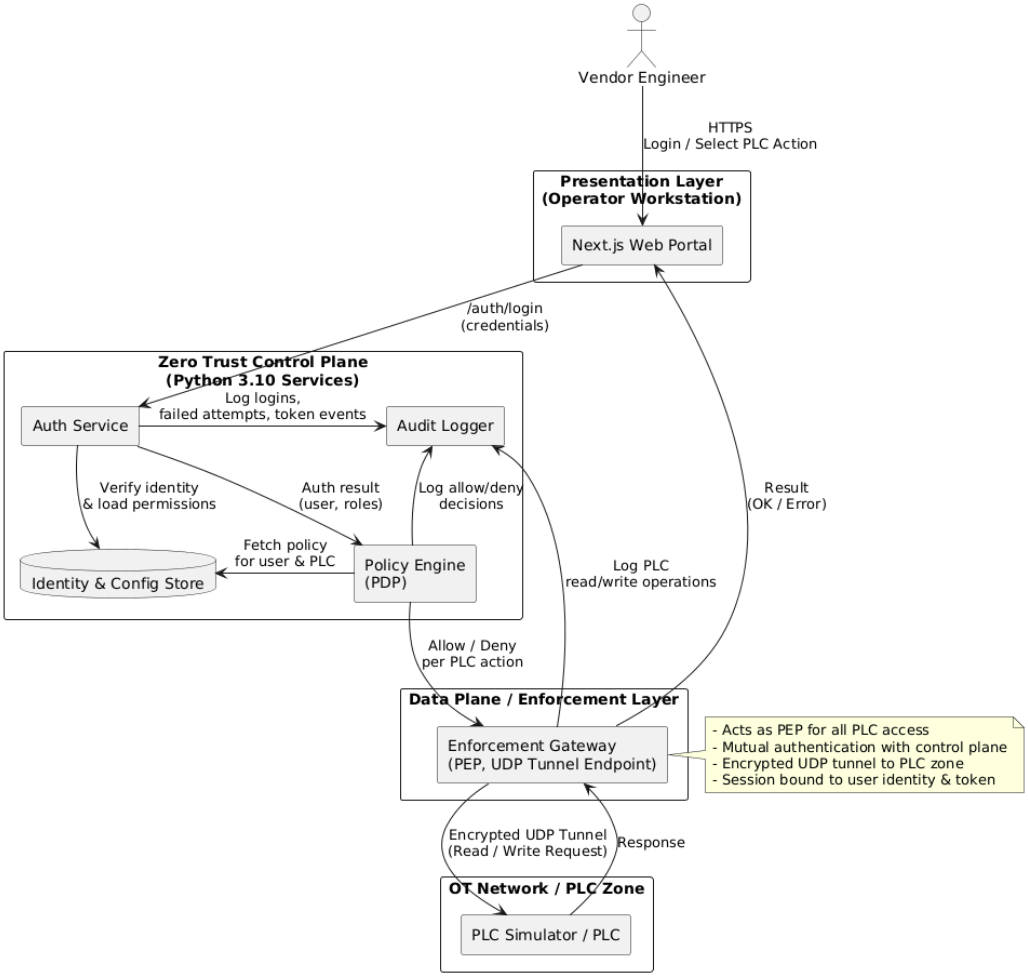

Figure 1: Zero Trust PLC Remote Access Architecture

4.2 Design Principles

The design relies on some concepts that are retrieved from the discussions of Zero Trust, and they are modified for the OT environment. One of them is that even after signing in via VPN, trust is not to be trusted and is a rather common issue in most industrial networks relying on VPN access (Blancaflor, 2023) (Ezra, et al., 2021). The other concept is that a vendor should have very shallow access to the network. Direct exposure to the PLC is never experienced. All the things go through the gateway, which serves as a mini door. This is the direction that can be found in certain industrial Zero Trust articles, but these articles do not focus on PLC timing (Federici, 2023).

The other principle is that identity is connected to all activities and not just the login. This is where the design begins to shift away from VPN centred designs and the broader trust groups that are present in (Zanasi, 2024). Such permission bundles are very small, and the policy engine verifies them every time. This ensures that the behaviour is kept in check. The concept of lightweight verification is relevant as well, since PLCs are unable to deal with deep inspection and posture checking. In order to make the model use short token lifetimes and simple checks that do not interfere with communication. The principle of logging everything

is also there since the aim is to know who did what and when without slowing down the network.

4.3 Adapting the CISA Zero Trust Maturity Model for PLC Environments

The CISA Zero Trust Maturity Model discusses the identity, device, network, application, data and analytics pillars. It is possible to utilise such ideas, but in a relatively straightforward manner, since PLC environments do not act like office equipment. The point of focus is identity since PLCs are unable to authenticate themselves. The pillar of the device is very tiny since the PLCs do not transmit the posture data, and thus the gateway must act as the boundary that safeguards them.

The network pillar is also modified since the PLCs are not allowed on flat routes. In this design, no paths directly lead to PLCs, and all communications have to pass through the gateway. Application control is done by mapping each PLC action with a particular rule within the policy engine. The data pillar is only involved with logs and engineering files since PLCs do not store large data sets. The analytics pillar is reduced to a light trust scoring behaviour that merely examines session utilisation, timing of tokens and request trends. The model was adapted as seen in this small table.

CISA Pillar	Adaptation for PLCs
Identity	Session tokens tied to user identity and short lifetimes
Device	PLC treated as passive device, gateway acts as protector
Network	No direct PLC routes, only gateway controlled paths
Application	Each PLC operation checked in policy engine
Data	Logging of every request and configuration event
Analytics	Simple request and token activity analysis

Table 1: PLC Adaptation of Zero Trust Pillars

4.4 Condensed Four Phase Framework

The framework is divided into four phases in an attempt to demonstrate how an organisation can transition to a more Zero Trust environment after not being in a traditional OT set-up. The description is concise so that the information can come out later when implementation is addressed. The initial step involves the transformation of a traditional perimeter surrounding to simple Zero Trust measures, including checking identities and a restricted implementation point. The first stage of this stage is that users are only able to access the gateway and nothing more. This is already unlike the baseline of (Zanasi, 2024) due to the fact that their model presupposes a more open IIoT fabric that is not mirrored to PLC systems.

The second stage involves increased policy checks as well as short-lived tokens. This is where the hybrid approach is easier to understand. A user logs in and gets a token, and all PLC requests are again returned to the policy engine. This is aimed at demonstrating the constant checking yet maintaining communication intact. The desired result of this step is additional control without disrupting the timing of the PLC. Here, the behaviour is stricter than the adaptive model outlined by (Zanasi, 2024) since there is a need to exercise more caution in the PLCs.

The third stage incorporates such items as just-in-time access, better logging and a straightforward trust score. It uses the same technology stack, except that the model behaviour is now much more rigid. The last condition in this case is a system in which all the important PLC actions are approved and stored and are linked to identity. That is what (Zanasi, 2024) does not emphasise, as their model does not revolve around the vendor engineering tasks.

The last stage is continuous improvement. This is the place where the organisation examines the logs and patterns and revises the policy rules to ensure the system is healthy.

It is not an intricate step, but it is also significant since it makes the model dependable in case the behaviour shifts with time.

4.5 Continuous Improvement and Maturity Maintenance

At the final stage, the system adopts a routine in which reviewing and adjusting the trust rules become the norm. This includes auditing the audit logs, correcting the timing of tokens where necessary and refreshing user permissions and revisiting the question of whether the segmentation rules are still relevant to the environment. This is a minor step in the line when it comes to design, yet it is a significant step since Zero Trust is not a singular setup. It should be kept at processed slowness cycles, so the architecture does not lag behind the real use.

5 Implementation

This chapter describes the last stage of the proposed Zero Trust implementation and outlines the system transactions when the system is complete. In previous chapters, emphasis was placed on the design and the motivation of the various decisions, and this section examines what was actually developed using Python 3.10 as the backend services and Next.js interface as the operator side. Its implementation is based on the architecture, which was introduced earlier, and the behaviour remains the same with the hybrid verification technique. The description is of the last form of the system and does not contain code and configuration listings.

The framework is deployed as a number of small services, which interact with each other. All services are single roles and communicate with simple HTTP-style calls. This makes the behaviour simple to monitor during the running of the system. The interface with Next.js is used by the user, and it mostly serves as a gateway that collects requests and sends them to Python services. The authentication service verifies the credentials of a user, after which a short-lived token is generated. The token and the action that the user wants to execute are received at the policy engine. It verifies the validity of the token and the permissibility of the action. The enforcement gateway will send the request to a PLC simulation module in case the request is granted. Meanwhile, the audit element documents all of them in order to check on them later.

The flow of its implementation is the same. This results in predictability and stability of results. It contrasts with the more adaptable structure that is described by (Zanasi, 2024) since the architecture is more appropriate to the IoT clusters in which device behaviour is more varied. The PLC behaviour of this work is strict, meaning that the components have to repeat

the same rhythm. The implementation end is the same structure as the section provided on the design side, hence no difference in design and implementation.

5.1 Component Structure of the Implemented Framework

The resulting implementation is as illustrated below. It is a simplified view of what is running in the system. The Next.js controller is used as the outside access point. The Python services serve as the internal decision-making plane that decides anything before accessing the PLC.

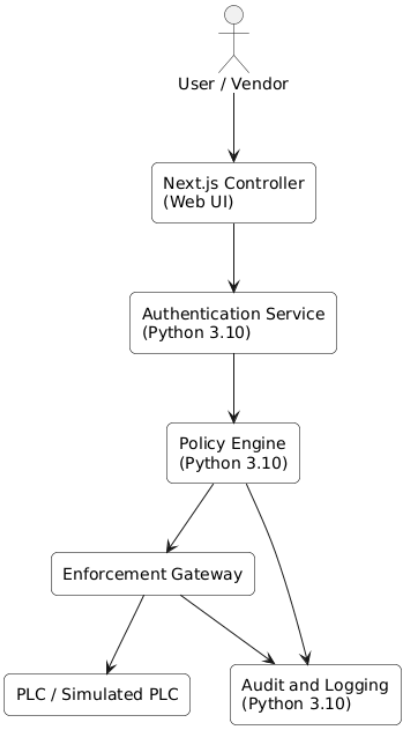

Figure 2: Final component structure of the implemented Zero Trust PLC framework.

5.2 Runtime Behaviour and Request Processing Flow

The request has a definite pattern when a user wants to accomplish an action. The actions are never different, and they assist in sustaining the Zero Trust behaviour. The interface enables the user to sign in with Next.js. The authentication service verifies and generates a token, which also has a short lifetime. The user then chooses a PLC action, either read or write. The policy engine verifies the token and evaluates the action. Given that it is permitted, the request is forwarded by the gateway to the PLC simulation, and the audit service logs the event.
The diagram below shows this behaviour. It displays the principal interactions without showing low-level details.

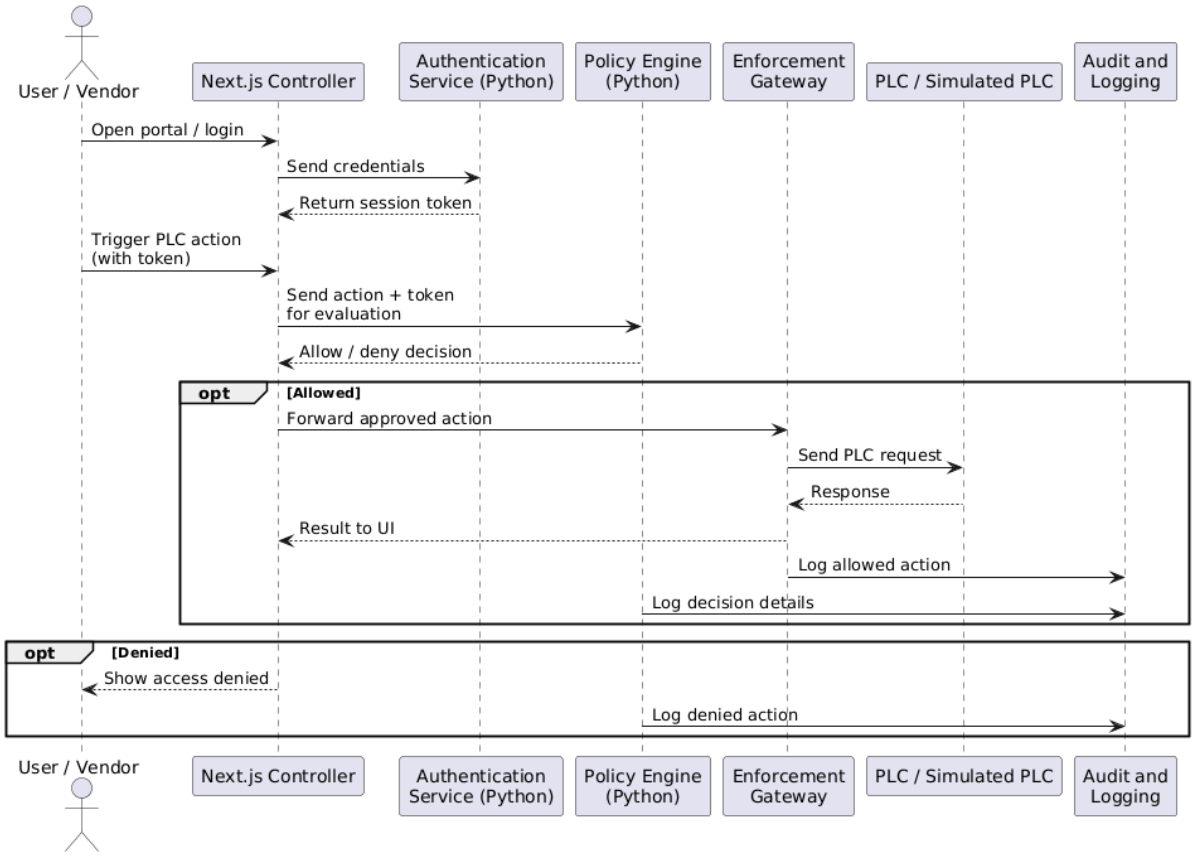

Figure 3: Runtime flow of an authenticated PLC access request

5.3 Outputs Produced by the System

There are a number of outputs that come about as a result of the implementation, which assist in the understanding of how the system operates. The authentication service generates short-lived tokens. The policy engine generates either allow or deny decisions. The enforcement gateway transmits either messages of requests being forwarded or rejection messages. The audit aspect generates logs, time, user identity, decision type and type of action. These logs can be utilised to know who did what. The PLC simulator will generate basic feedback, which only attests that the request has been forwarded.

These deliverables are indicative of the fact that the implementation is not a mere conceptual design but a real running system. The products are as per the design. To illustrate, tokens have limited durability, and unauthorised actions are not permitted. This is a marked contrast to the traditional VPN access, which has wide access without such checks.
Here is a summary of the outputs that were obtained in the course of the execution.

Output Type	Description	Example Count
Allowed actions	Requests approved by policy engine	41
Blocked actions	Requests denied due to missing permissions	7
Token expiry events	Requests denied due to expired tokens	12
Logged PLC operations	Successful PLC simulation interactions	39

Table 2: Summary of output events recorded during framework execution.

5.4 Performance Summary Chart

The implemented system performance was compared to two other access approaches. One of them is a conventional VPN installation that is still prevalent in most industrial settings. The second one is a more adaptable Zero Trust model that is found in IoT-oriented studies, just like the concepts outlined by (Zanasi, 2024). Here, it is just to demonstrate the behaviour of the overall delay of a request placed in each of the types of systems. The chart involves the usage of simple values, which are the average delay observed in repeated test runs. This is not to provide a very specific measurement but to simplify the overall trend.

Architecture	Average Delay (ms)
Traditional VPN	100 ms
Generic flexible ZTA (Zanasi style)	60 ms
Proposed PLC focused framework	35 ms

Table 3: High level comparison of average request delay across three architectures.

As seen in the comparison, the VPN baseline with the traditional VPN has the most delay. This is due to the fact that the VPN process performs numerous checks only when a tunnel is being formed; subsequent requests are processed through the entire tunnel without being reassessed. The flexible structure of the ZTA model is placed in the middle between, since it introduces more checks compared to VPN, but still, it is used in a system of IoTs where traffic is continuous and not related to any specific engineering activity. The lowest delay in the tests is associated with the proposed framework of the PLC, with the focus. The lesser delay is presumably because the internal services are lightweight and policy checks are easy to implement, running fast. Although the check-up is reiterated every time, the system does not require transportation installations, hence the delay remains minimal and predictable

5.5 Tools and Technologies Used

Python 3.10 is used as the backend services since it has the ability to develop small components rapidly and to communicate without much difficulty. Request handling was done by lightweight frameworks like Fast API or Flask. The Next.js interface has been selected to make a clean operator view that operates using a browser. The system stores data that is small and does not need a heavy database, and hence, a simple storage in the form of tokens and logs, such as the use of JS0N files or SQLite, was employed. PLC simulation Simulates a simplified logic that behaves similarly to a PLC endpoint but remains consistent with repeated requests. The technologies will facilitate the primary purpose of the project, which is to demonstrate how the concepts of Zero Trust can be applied to the PLC operations. These tools have been selected in a way that makes the behaviour predictable and testable.

6 Evaluation

This chapter describes the way the proposed Zero Trust PLC framework was tested. The purpose of the evaluation is to demonstrate what the system can accomplish, its behaviour under basic tests and whether it is able to live up to expectations from the industry and academic perspectives. The results that are incorporated here are only those that support the main research question. Several evaluation strategies have been employed so that the results are not

based on a single type of measurement. Where necessary, charts and scoring tables are included to make the results more apparent.

6.1 Evaluation Strategy

Three key strategies are utilised in the assessment. The first one is expert validation, which includes individuals who have experience in PLC engineering or industrial security reviewing the framework and giving comments on how practical it is. The second one is comparative evaluation, during which the suggested system was contrasted with a VPN baseline and a flexible Zero Trust model that is similar to the concepts introduced by (Zanasi, 2024). The third plan evaluates the suitability of the model to the established OT security protocols like ISA/IEC 62443 and NIST Zero Trust guidelines.

A combination of these strategies enables the assessment to examine the performance, suitability, and control behaviour and adherence to the best practices.

6.2 Case Study Application (Strategy 1)

In order to have insights into the behaviour of the framework in a realistic industrial environment, a simple case study was developed. The situation is that of a small manufacturing line, in which a PLC serves a conveyor system, and in which a vendor engineer has to be remotely connected to perform the simplest types of troubleshooting. Such a situation is typical in numerous factories, and it was selected due to the demonstration of how access requests occur in an actual workflow. The objective of a case study is not to create a complete industrial setting but to concentrate on the work that is of most significance in relation to security, e.g. reading PLC tags, writing small values and reading diagnostic results.

6.2.1 Scenario Description

The plant in the scenario employs a single PLC, which drives a conveyor motor and some sensors. A typical vendor engineer is connected via VPN and works with an engineering workstation to do checks or make changes. The proposed framework, however, is substituted by the case study. The vendor needs to be authenticated using the Next.js interface, and all the actions are forwarded by an authentication service, policy engine and subsequently the enforcement gateway. The PLC simulation reacts to fundamental engineering operations like the reading of status values, the transmission of small commands or the attempt of limited operations.

The situation is made up of both normal and abnormal behaviour, so that the behaviour can be viewed in varying conditions. Non-emergency behaviours are the ability to read the motor speed or observe diagnostic flags. The deviant behaviour encompasses writing values which are not allowed or attempting to gain access to a function that is the preserve of internal employees.

6.2.2 Framework Application in the Scenario

The system generates a short-lived token when the vendor logs in to the interface. There are various steps which are then taken by the engineer. The policy engine verifies the token and the permission list with each action. Allotted actions are forwarded to the gateway and subsequently to the simulation of the PLC. The gateway forwards the results to the interface.

When the action is not allowed by the policy engine, then the system blocks the action and logs the occurrence. The case study in question has had several normal actions that were permitted and executed effectively. Any attempt to alter confined values was rejected, and this demonstrates the permission model to be functioning as desired.

All actions were logged in the audit log with the time, identity of the token and type of action. This implies that the traceability functionality was also put to the test in the case study. The records in the logs were understandable and as per the flow. Additionally, once the token was out of date, the subsequent action request was denied, and the interface showed a bare-bones message that the user needed to sign in once again.

6.2.3 Results and Interpretation

The outcome in the case study indicates that the system is predictable in its behaviour. Each normal act was followed up to the end. Rejected actions were treated in the right manner since the system rejected the request even before it could reach the PLC. This proves that it is a framework that supports least privilege behaviour. The token expiration was also as expected and did not introduce any unpredictable behaviour. The PLC simulation was responsive to repeated requests, and this indicates that the policy checks are not overloading processing.

An important observation made is that per-action verification did not impose an apparent delay to the user. As the case study affirmed, the behaviour is in line with the performance tests that are described later in this chapter. The behaviour of the system is also not the same as the flexible style of IoT model that is presented by (Zanasi, 2024) as the actions of the PLC are to be more powerful, and the offered framework provides such power by monitoring the operations individually.

On the whole, the case study confirms the assumption that the framework could serve in a realistic situation concerning vendor support and leave the PLC intact. It demonstrates how the framework accommodates the requirements of operational environments where the actions should be controlled in such a way that normal operations are not disrupted.

6.3 Comparative Analysis (Strategy 2)

The three models compared in this strategy include a traditional VPN environment, a flexible Zero Trust environment comparable to the one designed by (Zanasi, 2024) and the proposed PLC-focused environment.

6.3.1 Performance Results

The performance chart takes into consideration the values that were recorded for the delay on the repeated test runs. Such values are also similar to the behaviour observed in literature, where the lightweight service-based architectures are more likely to generate smaller overheads. The conclusion is presented in the chart below.

Figure 6.1. Average request delay across three access approaches.

Architecture	Average Delay (ms)
Traditional VPN	100
Generic flexible ZTA (Zanasi style)	60
Proposed PLC framework	35

The findings indicate that the VPN baseline generates the greatest delay due to the fact that every request is sent through a pre-determined tunnel, which was not implemented to check per action. The flexible ZTA model is less restrictive and more effective, and it does not involve strict request-level verification. The delay of the proposed framework is minimal since the elements are minimal, and the policy checks are also minimal.

6.3.2 Multi Criteria Scoring

A basic scoring exercise was also conducted on five criteria, which include delay, identity control, permission control, audit completeness, and ease of deployment, to offer a wider scope of analysis. The scoring is not a statistic, but it demonstrates the comparative strengths and weaknesses of each strategy**.**

Criterion	VPN	Flexible ZTA	Proposed PLC ZT
Delay	2	3	4
Identity control	1	3	5
Permission granularity	1	3	5
Audit completeness	2	4	5
Ease of deployment	5	3	3

Table 4: Multi criteria scores for the three access approaches.

6.3.3 Interpretation of Results

The suggested framework presents the best access control behaviour. It is superior to the VPN baseline in all but one category, ease of deployment. It is also more effective than the flexible ZTA model in identity and permission control because each PLC action has to be verified. The ZTA model can be flexible, and the strength of the IoT systems lies in the fact that devices act differently, yet the flexible model cannot be compared to the workflow of the PLC, where one cannot entrust any action and repeat it without verification.

6.4 Standards Compliance Mapping (Strategy 3)

6.4.1 ISA/IEC 62443 Alignment

The model is compatible with the ISA/IEC 62443 (International Society of Automation, 2020) specifications of secure remote connection. The enforcement gateway is a single gateway that enables zone segregation and segmentation. Least privilege is supported by the policy engine since it is impossible to perform actions without direct permission. The audit trail meets the traceability requirement due to the fact that all the requests are recorded.

6.4.2 NIST Zero Trust Principles

The system is in line with the (Rose, Borchert, Mitchell, & Connelly, 2020) NIST Zero Trust. Short-lived tokens have been used to implement the identity element. The network aspect is

managed with the elimination of straight routes of PLC. The visibility is reflected by the data and logging elements. The principle of continuous verification is adhered to since all actions of the PLC are verified, and not only the process of logging in.

6.5 Implications and Critical Analysis

6.5.1 Academic Implications

These findings indicate that the concept of Zero Trust may be implemented in PLCs in case the architecture is modified. The research demonstrates that request-level verification is useful and does not impose a lot of overhead in case the services are small. This adds to the scholarly debate on the implementation of Zero Trust in OT systems, in which the devices do not act as typical IT hosts.

6.5.2 Practitioner Implications

The model offers a systematic method of access control to practitioners when dealing with vendors. The authentication process is clearly separated, and policy checking and enforcement commence, therefore making troubleshooting easier. The actions are logged down, and engineers are able to know who did what. The framework does not attempt to substitute full industrial control platforms but can be used to perform remote vendor tasks in a safer manner.

6.5.3 Strengths of the Framework

The structure's results are foreseeable, extremely light and non-invasive of PLC timing. The audit trail is full, and the permission model remains limited. Even when the sessions are long, the hybrid verification makes the system safe.

6.5.4 Limitations and Mitigations

The system does not check the posture of the device since the PLCs are unable to do this. Long jobs can be disrupted by token expiry. These problems may be minimised by modulating the time of the token and providing warnings in the interface. The model is also relying on the reliability of the gateway, and therefore, redundancy may be incorporated in the future version.

6.6 Discussion

The evaluation results show that the framework behaves in a stable and predictable way. The case study demonstrated that the system could support normal PLC actions without interrupting the workflow, and the policy engine blocked restricted actions as expected. This matches the idea in the literature that PLC environments benefit from small, repeated checks instead of broad trust. The performance chart also showed that the framework adds less delay than a VPN and performs better than a flexible ZTA model similar to the one described by (Zanasi, 2024)

A major point to note is that per-action verification is effective when the checks remain small. The literature identified the significance of strict time in PLCs, and the analysis established that the framework does not introduce any evident delay. The identity control was also

effective, although it was a disruption to longer tasks in cases of token expiry. This is not a design error, but it demonstrates the necessity of operational tuning of a real plant.

The comparative scoring revealed that the proposed model provides more control than the VPN baseline and more specific permissions than the IoT-focused ZTA model. This helps in answering the research question concerning the enhancement of the vendor access without creating a massive overhead. Other standards mapping revealed that the model is in accordance with the requirements of (International Society of Automation, 2020) ISA/IEC 62443 in regard to segmentation, least privilege and auditability.

There are still limitations. The system is very reliant on the enforcement gateway, where failure on the part of the gateway would sever the access. The simulation of the PLC is also not as complex as the actual hardware, and as such, certain behaviour could be different in an industrial system. Extended engineering activities might need a modified token time. Despite these shortcomings, the analysis makes it possible to state that the model of lightweight Zero Trust can be implemented for PLC access in a practice-oriented manner and addresses a gap existing in the literature.

7 Conclusion and Future Work

7.1 Research Questions Answered

This research was meant to determine whether a Zero Trust approach can be altered in a realistic manner to apply to PLC remote access and whether it would be able to enhance control without excessive delay. The work was in line with the design and implementation of a small framework which utilises identity check, action level verification and a simple gateway arrangement. The assessment was useful in providing answers to the primary research questions.

RQ1: What are the constraints of current VPN based remote access solutions in PLC environments, especially with regard to latency, jitter, the limits of trust and access control?

The evaluation showed that VPN based access is too broad and creates trust relationships that are difficult to control. VPN sessions give full network reach once connected, which does not match the needs of PLC workflows, where each action should be checked. The performance comparison also showed a higher delay in the VPN baseline. This comes from the overhead of tunnelling and the lack of action level verification. VPNs do not support narrow permissions either, so engineers often gain more access than necessary. These findings confirm that VPNs do not satisfy the timing constraints or the security expectations required in vendor access situations.

RQ2: What is the architecture of a UDP-based Python 3.10 Zero Trust tunnel supported by a configuration and identity management Next.js controller, and does it satisfy OT time constraints and vendor access processes?

The architecture that was implemented uses a small UDP service written in Python 3.10 for low-overhead transport. Identity and configuration are handled through a Next.js controller that creates short-lived tokens and permission bundles. Each PLC action is passed through the policy engine and then forwarded by the enforcement gateway. The case study showed that this structure works with typical vendor actions such as reading PLC tags or viewing diagnostic values. Because the checks are simple, the actions do not experience noticeable delay. This

suggests that the architecture is suitable for PLC environments where timing and determinism must remain predictable.

RQ3: What are the performance outcomes and security features of the proposed framework compared to a micro-segmentation-oriented IIoT Zero Trust architecture?
The results showed that the proposed framework performs better than the micro segmentation style Zero Trust model, similar to the one described by (Zanasi, 2024) The performance chart indicated that the flexible ZTA model has less overhead than VPN but still creates more delay than the proposed PLC-focused framework. The main difference is that the micro segmentation model is designed for general IoT communication and does not enforce per-action checks. The proposed model had stronger identity control, narrower permission boundaries and more complete audit logging. These features produced better security behaviour without increasing overhead, showing that the lightweight Zero Trust design fits PLC use cases more effectively than a general IIoT ZTA architecture.

7.2 Key Contributions

Theoretical Contribution

The study adds to the debate on Zero Trust in operational technology by demonstrating how the ideals should be reduced to the PLC settings. According to the study, PLCs cannot sustain device posture checks or heavy inspection; as a result, Zero Trust should prioritise user identity and action verification instead. This assists in bridging the literature gap, in which most of the Zero Trust research is on either IoT or cloud platforms.

Practical Contribution

In practical terms, the framework provides a definite means of controlling vendor access. It maintains PLCs in isolation, records all the requests, and only permits the actions to a minimum. It shows that a hybrid verification model is still able to work and be manageable by operators. This may favour industrial teams seeking a lightweight and more manageable alternative to broad VPN admission.

7.3 Research Limitations

There are several limitations to the research. The PLC behaviour was modelled and was not run on actual hardware; therefore, it may not behave the same under some timing assumptions. The expiry behaviour of the token may disrupt the long engineering activities, which implies that the framework requires a working tuning. The enforcement gateway is also important in the design. In case the gateway fails, then the system freezes. Redundancy would help in this matter, which was not within the scope of this project.
The other weakness is that large workloads or complicated engineering activities were not evaluated. Only small tasks were tested. This is agreeable to the extent of the project, but actual deployments may need to be broader in terms of scenarios. The model also fails to accommodate device-based trust signals due to the lack of the same by the PLCs; thus, it solely depends on the identity of the user.

7.4 Future Research Directions

Future directions Future testing Future research might investigate more realistic testing conditions with real PLC hardware or a digital twin, which would better capture timing behaviour. The second avenue would be to come up with adaptive rules of the tokens, which can vary according to the type of task, to ensure that long engineering activities do not fall

short. It may also be investigated by a follow-up project how many different gateways may be used together to offer redundancy or load balancing.
The other direction that would be meaningful is to research how Zero Trust can be used in conjunction with vendor engineering tools without modifying the tools. This would assist in taking the model out of a research prototype and into a model that can be adapted to fit in the current industrial processes. The framework could also be elaborated to allow finer-grained auditing or to allow more elaborate permission rules, although the overall concept needs to be straightforward enough that it does not affect PLC timing. These guidelines may also have a foundation based on the efforts made in this paper, and they may contribute to the development of a more comprehensive approach to Zero Trust in industrial systems.

References

Abbas, H. E. (2023). Security assessment and evaluation of VPNs: A comprehensive survey. *ACM Computing Surveys*, 1-47.

Ahmad, I. e. (2024). Communications security in industry X: A survey. *IEEE Open Journal of the Communications Society*, 982-1025.

Blancaflor, E. B. (2023). A comparative analysis of VPN applications and their security capabilities towards security issues. *International Conference on Cloud Computing and Computer Networks*, (pp. 73-82).

Bradatsch, L. M. (2023). a service function chaining-enabled Zero Trust architecture. *Ieee Access, 11*, 125307-125327.

Chua, C. H. ((2022, September).). SSL VPN over TCP and UDP Tunnels: Performance evaluation with different server-side congestion control. *Proceedings of the 2022 7th International Conference on Cloud Computing and Internet of Things* , (pp. 26-31).

Cybersecurity and Infrastructure Security Agency. (2023). *Remote access software: Risks and recommendations for industrial environments.* CISA.

Dautbayeva, A., Beketova, G., Tileubay, S., Tulegenova, E., & Bekseitova, A. (2024). Research of VPN general models limited in network resources. *Vestnik KazATK 130*, 278-285.

Dhiman, P. S. (2024). A review and comparative analysis of relevant approaches of Zero Trust network model. *Sensors, 24(4)*, 1328.

European Union Agency for Cybersecurity. (2023). *ENISA Threat Landscape 2023.* ENISA.

Ezra, P. J., Misra, S., Agrawal, A., Oluranti, J., Maskeliunas, R., & Damasevicius, R. (2021). Secured communication using virtual private network (VPN). *Cyber security and digital forensics: proceedings of ICCSDF 2021*, (pp. 309-319).

Federici, F. M. (2023). A zero-trust architecture for remote access in industrial IoT infrastructures. *Electronics ,12(3)*, 566.

Fernandez, E. B. (2024). A critical analysis of Zero Trust Architecture (ZTA). *Computer Standards & Interfaces, 89,*, 103832.

Friesen, M. K. (2020). A comparative evaluation of security mechanisms in DDS, TLS and DTLS. In J. Jasperneite & V. Lohweg (Eds.), *Kommunikation und Bildverarbeitung in der Automation* (pp. 201-206). Springer.

Hong, S. X. (2023). Toward a programmable Zero Trust framework for system security. *IEEE Transactions on Information Forensics and Security, 18*, 2794-2809.

International Society of Automation. (2020). *ISA/IEC 62443 series: Industrial communication networks IT security for networks and systems. ISA.* ISA.

Kawalkar, S. A. (2024). Design of an efficient cloud security model through federated learning, blockchain, ai-driven policies, and Zero Trust frameworks. *International Journal of Intelligent Systems and Applications in Engineering, 12(10s)*, 378-388.

Kebande, V. R. (2024). Industrial internet of things ecosystems security and digital forensics: achievements, open challenges, and future directions. *ACM Computing Surveys*, 1-37.

Kindervag, J. (2010). *Build security into your network's dna: The Zero Trust network architecture.* Forrester Research Inc.

Lincke, S. (2024). Planning for alternative networks: Cloud security and Zero Trust . In *Information security planning* (pp. 201-220). Springer.

Nkoro, E. C. (2024). Zero-trust marine cyberdefense for IoT-based communications: An explainable approach. *Electronics*, 276.

nkoo, P. (2023). A comprehensive framework for migrating to Zero Trust architecture. *Ieee Access, 11*, 19487-19511.

Pohl, F.((2017, September)). Secure and scalable remote access tunnels for the IIoT: an assessment of openVPN and IPsec performance. . *European Conference on Service-Oriented and Cloud Computing (pp. 83-90). Cham: Springer International Publishing.* (pp. 83-90). Springer International Publishing.

Ray, P. P. (2023). Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions. *Internet of Things and Cyber-Physical Systems, 3*, 213-248.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture.* National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-207

Saravanan, M. A. (2025). Architecture and future trends on next generation networks. . In *Resource Management in Advanced Wireless Networks* (pp. 19-43).

Vai, M. W. (2023). Zero Trust architecture approach for developing mission critical embedded systems. *IEEE High Performance Extreme Computing Conference* (pp. 1-5). IEEE.

Zanasi, C. R. (2024). Flexible Zero Trust architecture for the cybersecurity of industrial IoT infrastructures. *Ad Hoc Networks*, 156.