

MIDOCEAN UNIVERSITY

Research Methodology in Computer Science (AI800)

Research Title

**Securing Cloud-Native Infrastructure in Telco-Cloud: Container
Isolation, Kubernetes Control-Plane Risk, and Multitenancy**

A Hardening Framework for the Shared Kubernetes Control Plane in Multitenant Telco CaaS

Ahmed Shawky Hewidy

Student ID: 253001655

Lecturer: Dr. Mohammed Anbar

Abstract

Telecommunications operators are increasingly deploying cloud-native network functions on shared Kubernetes-based Containers-as-a-Service (CaaS) platforms. Although shared clusters can improve resource utilization at the network edge, they also create a common control-plane boundary between tenants. Recent research has identified weaknesses in Kubernetes control-plane interfaces, including insufficient access restrictions and missing rate-limiting mechanisms, with potential consequences such as information leakage, control-plane resource exhaustion, and service degradation for co-located workloads.

This research proposes a layered security framework for multitenant telco CaaS environments hosting containerized 5G core functions, including the AMF, SMF, and UPF. The framework will combine tenant-scoped access control, admission policies, API request management, and workload isolation using mechanisms such as Kata Containers and node-level separation. A controlled Kubernetes testbed will be used to compare a compliance-based baseline with the proposed framework. Documented control-plane attack scenarios will be replayed in the laboratory, and the configurations will be evaluated using information leakage, control-plane CPU utilization, co-tenant packet loss, legitimate workload performance, and resource overhead as metrics.

The expected outcome is an experimentally evaluated hardening framework that clarifies the relationship between tenant isolation, control-plane protection, and resource efficiency. The study will not claim to provide complete protection against all Kubernetes or 5G attacks; instead, it will provide comparative evidence for the attack scenarios and testbed conditions defined in the research scope.

Keywords: Telco-Cloud, Kubernetes, Containers-as-a-Service, Multitenancy, Cloud-Native Network Functions, Zero Trust, Container Isolation, 5G Security

Chapter 1: Introduction

Telecommunications infrastructure is being rebuilt on the same technology stack as the public cloud. Network functions that once ran as dedicated appliances, or later as virtual machines under Network Function Virtualization (NFV), are now packaged as containers and orchestrated by Kubernetes-based Containers-as-a-Service (CaaS) platforms. This shift is driven by cost and agility: a CaaS platform lets an operator deploy, scale, and update 5G network functions with the same automation used by hyperscale cloud providers, and it lets multiple tenants share one physical cluster rather than paying for dedicated infrastructure at every site (Şenel et al., 2023). At the network edge in particular, where compute and power are constrained, a dedicated cluster or even a dedicated virtual machine per tenant is often not economically viable (Aiello et al., 2026; Şenel et al., 2023).

That efficiency comes with a structural change in the attack surface. In a Kubernetes cluster, every tenant's workload ultimately talks to one control plane, the API server, etcd, the scheduler, and the admission and access-control layers that decide what any container is allowed to do. When that control plane is shared, it becomes a single trust boundary carrying every tenant's traffic and every tenant's risk at once. Wang et al. (2025) showed empirically, on Alibaba Cloud and Taikun Cloud, that this boundary does not hold under attack: a malicious container reachable through routine tenant privileges was able to leak sensitive cluster information, push control-plane CPU utilization to 90%, and cause a co-located victim tenant to lose up to 70% of its packets. Because the tenants sharing that control plane in a telco CaaS deployment are 3GPP network functions such as the AMF, SMF, and UPF, a breach here is not confined to one customer's application, it can degrade or expose the signalling and user-plane traffic of a live 5G network.

1.1 Background

1.1.1 From NFV to Cloud-Native Telco Infrastructure

The telecom industry's first attempt at software-based infrastructure was Network Function Virtualization: running network functions as virtual machines under a hypervisor, coordinated by an MANO (Management and Orchestration) stack. Pattaranantakul et al. (2018) catalogued the threats specific to that model, hypervisor breakout, tampered VNF images, insecure orchestration interfaces, and argued that NFV's programmability, the same property that makes it useful, widens the attack surface because control becomes centralized and software-defined. Alnaim et al. (2022) later formalized these threats as reusable misuse patterns in a security reference architecture. Both papers reason in terms of hypervisors and virtual network functions, a framing that predates the shift to containers and Kubernetes role-based access control (RBAC) that now defines telco-cloud deployments.

1.1.2 CaaS, Kubernetes, and Cloud-Native Network Functions

Cloud-native network functions (CNFs) replace VNFs with containerized workloads managed by Kubernetes, and CaaS platforms such as Şenel et al.'s (2023) EdgeNet demonstrate that a single shared cluster, combined with Kata Containers for VM-grade isolation and node-level slicing for tenants that need stronger separation, can deliver multitenancy efficiently enough for resource-constrained edge sites. That efficiency argument is the economic reason telco operators are adopting CaaS. But it also means the security question shifts from "is the hypervisor secure" to "is the shared Kubernetes control plane secure," and Morić et al. (2025) show that operators can answer that question with a compliance checklist,

hardened API server, encrypted etcd, RBAC, image scanning, admission control, without ever testing whether the hardened cluster resists an active adversary.

1.1.3 The Control Plane as a Shared Trust Boundary

Aiello et al. (2026) map 5G Standalone vulnerabilities directly onto the 3GPP functions and interfaces that run as containers on these clusters, the AMF, SMF, UPF, the N2 and N4 reference points, and the service-based architecture APIs, and rank container escape and control-plane signalling floods among the highest-priority risks. Wang et al.'s (2025) attacks give that ranking empirical weight: the same control-plane interfaces every tenant depends on for orchestration are the interfaces a hostile tenant can abuse to attack every other tenant. Multitenancy, the very property that makes CaaS affordable at the edge, is what turns a single compromised container into a cross-tenant, cross-function incident.

1.2 Problem Statement

Telco clouds increasingly run 5G network functions as containers on shared Kubernetes clusters, and operators share one cluster among tenants because a dedicated cluster, or even a dedicated VM, per tenant is often not affordable at edge sites (Aiello et al., 2026; Şenel et al., 2023). Wang et al. (2025) found that the control-plane interfaces those tenants share suffer from two recurring weaknesses, insufficient access restriction and missing rate limiting, and demonstrated the consequences on two commercial clouds: leaked cluster information, control-plane CPU pushed to 90%, and a co-located victim tenant losing up to 70% of its packets.

This matters more in a telco setting than in a generic multi-tenant cloud, because of what runs on these clusters. The AMF, SMF, and UPF carry live 5G signalling and user traffic, and container escape is rated among the highest risks against them (Aiello et al., 2026). An attack that reaches the control plane reaches every tenant on that cluster, so one hostile workload can become a cross-tenant, cross-function problem rather than a contained incident.

The existing literature approaches this from three directions, and each stops short in a similar way. Pattaranantakul et al. (2018) and Alnaim et al. (2022) work at the architecture level but frame the problem in hypervisors and VNFs, not Kubernetes RBAC or container runtimes. Morić et al. (2025) harden the API server, etcd, and access control against compliance baselines, but a cluster can pass those checks and still be open to the exact interface attacks Wang et al. (2025) ran, because a configuration benchmark does not test for missing rate limits. Şenel et al. (2023) isolate tenant pods with Kata Containers and node slicing, which protects the workload but leaves the shared control plane itself untouched. The four additional studies in this review reinforce the same pattern from different angles: Alboqmi and Gamble (2025) show that trust between microservices in a service mesh has to be checked continuously rather than assumed once; El-Hajj (2025) shows Zero Trust cutting unauthorized access on the radio side of

O-RAN, but that work stops at the RIC and does not touch the Kubernetes control plane hosting the core network functions; and Liu et al. (2020) show that the container image supply chain feeding these clusters carries its own exploitable weaknesses that scanning alone does not close. None of the ten studies reviewed for this proposal has re-run Wang et al.'s attacks against a hardened design.

1.5 Research Objectives

The primary goal of this research is to secure the shared Kubernetes control plane in multitenant telco CaaS, so that tenant density does not come at the price of cross-tenant compromise. Five objectives serve this goal:

1. To identify and classify the Kubernetes control-plane interfaces a hosted container can reach in multitenant telco CaaS, mapping weaknesses of insufficient access restriction and missing rate limiting to containerized 5G network functions (Wang et al., 2025; Aiello et al., 2026). The planned output is a documented attack-surface taxonomy.
2. To design a control-plane hardening framework that adds per-tenant access restriction and rate limiting on top of existing compliance baselines, closing the weakness classes identified under the first objective (Morić et al., 2025; Wang et al., 2025).
3. To integrate the proposed framework with workload-level isolation, namely Kata Containers and node-level slicing, so that control-plane protection and container isolation operate together on one shared cluster (Şenel et al., 2023).
4. To evaluate the framework, on a lab-scale multitenant testbed, by replaying tenant-originated control-plane attacks and measuring information leakage, control-plane CPU load, and co-tenant packet loss, against a compliance-only hardened baseline (Wang et al., 2025; Morić et al., 2025).
5. To compare the resource cost of the validated defenses against the efficiency that motivates shared clusters in the first place, giving operators a measured basis for weighing tenant density against attack resistance at the edge (Şenel et al., 2023).

Each objective produces something checkable: a taxonomy, a framework design, an integration, a set of attack-replay measurements, and a cost comparison. The first objective defines the boundaries of the problem, the second and third build the defense, and the fourth and fifth are planned to deliver the adversarial and economic evidence that current work on multitenant telco CaaS does not provide.

1.6 Research Scope

This research is scoped to the Kubernetes control plane and its immediate workload-isolation layer, inside a multitenant telco CaaS deployment hosting containerized 5G Standalone core functions (AMF, SMF, UPF). In scope: control-plane interface analysis and taxonomy; design of per-tenant access-restriction and rate-limiting controls layered on Kubernetes RBAC, admission control, and a policy engine; integration with Kata Containers and node-level slicing; and a lab-scale adversarial evaluation using attack strategies already documented in the literature, run on a small multitenant cluster rather than production infrastructure.

Out of scope: physical and radio-layer security of the RAN or O-RAN interfaces, which is mentioned in Chapter 2 only to place Zero Trust work in context; non-Kubernetes orchestration platforms; a full compliance audit against 3GPP SA3 or ETSI NFV-SEC as a deliverable in its own right, although both inform

the framework's design; and production-scale or commercial deployment, which is left for future work beyond what this proposal can cover.

Chapter 2: Literature Review

This chapter extends the studies synthesis, each checked against its publisher record rather than taken from memory. The ten studies fall into four groups: NFV-era architecture, Kubernetes hardening and multitenant systems, empirical attacks and 5G vulnerability mapping, and three adjacent layers (service mesh trust, radio-side Zero Trust, and container supply-chain security).

2.1 Architectural and NFV-Era Foundations

Pattaranantakul et al. (2018) built a use-case-driven threat analysis of the ETSI NFV architecture. They catalogued hypervisor breakout, tampered VNF images, insecure MANO interfaces, and insider misuse from the orchestrator's position, and argued for a centralized security-orchestration layer rather than defending each function separately. Alnaim et al. (2022) took a similar threat space and expressed it as reusable misuse patterns in UML, aiming for something teams could actually instantiate instead of a one-off write-up. Neither paper is about containers. Both are useful for understanding how centralizing control widens an attack surface, but the vocabulary is hypervisors and VNFs, and it takes some translation before it says anything specific about Kubernetes RBAC or admission controllers.

2.2 Kubernetes Operational Hardening and Multitenant CaaS Systems

Morić et al. (2025) walk through Kubernetes architecture and then apply hands-on hardening and compliance testing to the API server, etcd, and access-control layer, checking configurations against recognized baselines. It reads as a genuinely useful hardening playbook. The limitation is one the authors do not claim to solve: passing a compliance check says nothing about whether the configuration holds up against someone actively trying to break it. Şenel et al. (2023) approach the problem from the workload side instead. Their EdgeNet framework, open source and running in production, lets tenants share one cluster while routing pods through Kata Containers for stronger isolation, or reserving whole nodes when a tenant needs more separation than a container can offer. It is a solid demonstration that shared multitenancy works technically and makes economic sense at the edge, but the shared control plane underneath it is never itself put under attack. The isolation is shown to function, not shown to survive an adversary.

2.3 Empirical Attacks and 5G-Specific Vulnerability Mapping

Wang et al. (2025) is the study that gives this proposal its evidence. Using static analysis of Kubernetes control-plane interfaces, the authors found two recurring flaws, insufficient access restriction and missing rate limiting, turned them into four attack strategies, and ran seven attacks across four case studies on Alibaba Cloud and Taikun Cloud. Four attacks succeeded on Alibaba, five on Taikun. The measured impact included leaked cluster information, control-plane CPU pushed to 90%, and packet loss up to 70% for a co-located victim tenant. Both vendors confirmed the findings and committed to fixes, which is a reasonable indicator the results were not an artifact of a misconfigured test environment. It is worth being precise about what the study does and doesn't establish: it demonstrates that these two clouds, with their specific default configurations, were vulnerable to these specific attacks. It is not a claim that every Kubernetes deployment fails the same way, and this proposal does not treat it as one.

Aiello et al. (2026) complement this with a PRISMA-based systematic review mapping 5G Standalone vulnerabilities onto 3GPP functions and interfaces: AMF, SMF, UPF, the N2 and N4 reference points, and service-based architecture APIs. Each vulnerability is tagged by evidence level, from theoretical to live-exploited, and control-plane signalling floods, PFCP misuse on N4, and container escape (including CVE-2024-21626, a SELinux verification flaw that grants full host access) come out as the highest-priority risks. Between the two papers, the problem in Chapter 1 stops being hypothetical: Wang et al. show the control plane is attackable in practice, and Aiello et al. show which 3GPP functions are sitting behind it when that happens.

2.4 Adjacent Security Layers: Service Mesh, Radio-Side Zero Trust, and Supply Chain

Three studies from outside the original six round out the layers a multitenant telco CaaS deployment actually depends on. Alboqmi and Gamble (2025) extend the Service Mesh Risk-Adaptive Access Control (SMAAC) model, scanning microservices continuously with Trivy and assigning CVSS-weighted trust scores inside an Istio service mesh, tested on the Train Ticket benchmark application. Their point is a narrow one but a fair one: trust between microservices decays as vulnerabilities are disclosed, so a static trust assumption made at deployment time goes stale. That applies to inter-CNF communication inside a shared cluster, though it is a different layer from the control-plane hardening this proposal focuses on, and the two are meant to sit alongside each other rather than substitute for one another.

El-Hajj (2025) proposes a Zero-Trust and federated-learning framework for O-RAN, enforcing continuous authentication and micro-segmentation on RAN Intelligent Controller (RIC) components, and reports an 89% drop in unauthorized RIC access attempts against a conventional O-RAN baseline. That number comes from simulation, not a production deployment, so it should be read as a promising result rather than a settled one. More importantly for this proposal, the scope is the radio side (the RIC and its xApps) and does not touch the Kubernetes control plane hosting the 5G core functions this proposal is about. The two are easy to conflate because both use the phrase "Zero Trust for telco," but they are securing different components.

Liu et al. (2020) is older than the rest of this review but still the most relevant large-scale study on the container supply chain: an analysis of 2.2 million Docker Hub images that found dangerous default run-command parameters, 42 confirmed malicious images capable of remote code execution or cryptomining, and widespread delayed or missing patching across the public registry. None of this happens at the control-plane layer Wang et al. attack, but it happens earlier in the pipeline that feeds that control plane. A hardened control plane can still schedule a compromised image if nothing upstream is checking what gets pulled into the cluster, which is why Chapter 3 treats registry and image provenance as a boundary condition for the framework rather than a separate research thread.

2.5 Critical Comparison

Put side by side, the ten studies split fairly cleanly by the kind of evidence behind them. Wang et al. (2025), Alboqmi and Gamble (2025), El-Hajj (2025), and Liu et al. (2020) all tested something, on production clouds, a benchmark application, a simulation, or a 2.2-million-image dataset respectively. Pattaranantakul et al. (2018) and Alnaim et al. (2022) build frameworks and taxonomies with no

measurement attached; that is not a flaw in what they set out to do, but it does mean their conclusions describe what could go wrong rather than what has been shown to go wrong. Morić et al. (2025) sits in between: the hardening steps are tested against a compliance baseline, but not against an attacker, which is exactly the distinction Wang et al.'s results turn out to matter for.

There is also a genuine disagreement in this set, not just a gap. Şenel et al. (2023) argue that a shared cluster with selective Kata isolation is efficient enough for the edge and safe enough for most workloads. Their EdgeNet deployment is the evidence. Wang et al.'s results cut against the default version of that claim, because the tenant isolation Şenel et al. describe does not touch the shared control plane, and that is exactly where Wang et al. found the damage. Both findings can be true at once. EdgeNet's workload isolation is not what Wang et al. attacked, and Wang et al.'s target is not what EdgeNet was built to solve. But it means a security case for shared multitenancy that leans only on Şenel et al. is incomplete, and this proposal is really trying to close that specific gap between the two.

A second split is about framing rather than evidence quality. Pattaranantakul et al. and Alnaim et al. think in hypervisors and VNFs. Şenel et al., Morić et al., and Wang et al. think in containers, runtimes, and Kubernetes objects. Aiello et al. bridge the two by mapping 3GPP functions onto containerized deployments. For a study about Kubernetes-hosted CNFs, the container framing is the one that transfers directly. The NFV-era papers still inform the threat vocabulary, but their specific countermeasures, hypervisor attestation, VNF image signing, need reinterpretation before they apply here, and this proposal does not assume that reinterpretation is automatic.

2.6 Synthesis Matrix

Study	Layer	Key Contribution	Adversarial Evidence?	Gap Relative to This Proposal
Pattaranantakul et al. (2018)	NFV architecture	Use-case-driven NFV threat taxonomy; security-orchestration layer	No	VM/hypervisor framing, not container/K8s
Alnaim et al. (2022)	NFV architecture	Pattern-based NFV security reference architecture (UML)	No	Abstract, unvalidated; not container-native
Şenel et al. (2023)	Workload / CaaS system	EdgeNet: multitenant CaaS via Kata Containers + node-level slicing	No (functional demo only)	Shared control plane left unaddressed
Morić et al. (2025)	Control plane (ops)	Hardening + compliance assessment of API server, etcd, access control	No (benchmark-based)	Compliant ≠ attack-resistant
Wang et al. (2025)	Control plane (attack)	7 attacks / 4 case studies on Alibaba & Taikun; up to 90% CPU, 70% packet loss	Yes (production clouds)	No hardened defense re-tested against it

Study	Layer	Key Contribution	Adversarial Evidence?	Gap Relative to This Proposal
Aiello et al. (2026)	5G core (3GPP mapping)	PRISMA review mapping 5G SA vulnerabilities to AMF/SMF/UPF, N2/N4	Mixed (evidence-tagged)	Review only; no new empirical data
Alboqmi & Gamble (2025)	Service mesh	Vulnerability-driven, runtime trust scoring in Istio service mesh	Yes (benchmark app)	Inter-CNF trust, not control-plane hardening
El-Hajj (2025)	O-RAN / RIC	Zero-Trust + federated learning; 89% fewer unauthorized RIC accesses	Yes (simulation)	Radio side only; not the shared K8s control plane
Liu et al. (2020)	Supply chain / registry	Large-scale Docker Hub risk study (2.2M images); malicious images, weak patching	Yes (large-scale empirical)	Registry-level; does not address runtime control plane

Reading across the matrix, the same pattern from Chapter 1 shows up again. The studies with real adversarial evidence at the control-plane layer are not the studies proposing a defense at that layer, and the studies that do propose a defense there have not been checked against that evidence. It is worth noting this is not unique to the control plane: service mesh trust, radio-side access, and image supply chains are each treated as their own tested sub-problem elsewhere in the literature. The control plane is arguably the one place where every tenant's risk actually converges, and it is also the one still defended mostly by argument rather than by test.

Chapter 3: Research Methodology

The methodology runs in four phases: a literature review, a qualitative phase that builds the taxonomy and the framework design, a quantitative phase that tests the design against a real attack, and a framework development and validation phase that pulls the first three together into one evaluated artifact.

3.1 Phase 1: Literature Review

Sources: IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink, plus standards documents from 3GPP (SA3 series), ETSI (NFV-SEC series), CNCF, and NIST (SP 800-204, SP 800-207). Search terms pair "Kubernetes," "container," "multitenant," or "CaaS" with "control plane," "5G," or "telco." A study is included if it is peer-reviewed or an official standards document, published within roughly the last eight years unless it is foundational work still cited as current, and speaks directly to container or Kubernetes-layer security, or to a 3GPP-mapped 5G vulnerability. Vendor blog posts, marketing material, and papers about application-layer security with no infrastructure component are excluded. Screening goes title and abstract first, then full text for anything that survives, then data extraction into the synthesis matrix in Chapter 2. The ten studies there are a starting point, not a final count. I expect the list to grow once Phase 2 exposes gaps the current search terms missed.

Analysis technique: thematic synthesis across the four groupings used in Chapter 2 (NFV-era architecture, Kubernetes hardening and multitenant systems, empirical attacks and 5G mapping, and the three adjacent layers), rather than a paper-by-paper summary. The output of this phase feeds directly into Phase 2.

3.2 Phase 2: Qualitative Research

This phase is document and artifact analysis, not interviews. It has two parts. The first classifies each control-plane interface Wang et al. (2025) documented as reachable (API server endpoints, admission webhooks, etcd access paths, scheduler interfaces) by which weakness class applies to it, insufficient access restriction, missing rate limiting, or both, and maps each interface to the 3GPP function whose confidentiality or availability it can affect, using Aiello et al.'s (2026) function-to-interface mapping as a starting template. That produces the attack-surface taxonomy.

The second part uses that taxonomy to design the hardening controls. The design adds per-tenant access restriction and rate limiting on top of the compliance baseline Morić et al. (2025) describe, using mechanisms already native to Kubernetes rather than custom tooling: RBAC scoped to a tenant's namespace, validating and mutating admission control through a policy engine (Open Policy Agent with Gatekeeper, or Kyverno; the choice depends on which integrates more cleanly once the testbed exists), and rate limiting at the API server via the built-in Priority and Fairness API, extended with per-tenant flow-schema quotas. This layer is then combined with Şenel et al.'s (2023) workload isolation, Kata Containers where a tenant needs VM-grade separation, node-level slicing where a tenant needs a dedicated set of nodes, so control-plane hardening and workload isolation are designed as one system rather than two

separate claims that happen to sit in the same proposal. Both parts of this phase are qualitative in the sense that matters here: the output is a classification and a design, not a number.

3.3 Phase 3: Quantitative Research

The testbed is a small multitenant cluster built with kind or minikube, most likely on a single reasonably capable workstation or a modest cloud VM rather than dedicated hardware, since this is a course project and not a lab with a rack of servers behind it. Real AMF, SMF, and UPF software is not something I expect to get access to, so the plan is to stand in with open-source 5G core implementations. Free5GC and Open5GS are the two candidates, running as containerized tenant workloads. This is a real limitation, not a detail to gloss over. An open-source core's resource footprint and behavior under load will not exactly match a commercial vendor's AMF, so any measured effect should be read as indicative of the control-plane mechanism, not as a validated number for a specific vendor's deployment.

Sample: Wang et al.'s (2025) four documented attack strategies, run against two configurations, the compliance-only baseline from Morić et al. (2025) unmodified, and the hardened design from Phase 2. Each attack is repeated multiple times per configuration, at least ten runs each, resources permitting, so the measurements are not a single lucky or unlucky trial. Data collection: automated logging of the same three metrics Wang et al. used, whether cluster information leaks, control-plane CPU utilization during the attack, and packet loss for a co-located tenant. Using their metrics rather than new ones is deliberate. It is the only way the hardened result is comparable to a published baseline rather than a number invented for this proposal. Data analysis: descriptive statistics (medians and ranges) plus a non-parametric test such as the Mann-Whitney U test, given the likely small sample size and no reason to assume the results are normally distributed.

The resource cost of the hardening also gets measured here. Additional CPU and memory consumed by admission control, rate limiting, and workload isolation combined will be recorded against the compliance-only baseline and set against the density gains Şenel et al. (2023) report for shared multitenancy. The result is a rough cost-per-security-gain picture, not a formal economic model. That is a reasonable ambition for a lab-scale testbed, and claiming more precision than the setup can support would not be honest.

3.4 Phase 4: Framework Development and Validation

Based on insights of Development the taxonomy from Phase 2, the hardening design from Phase 2, and the measured results from Phase 3 are combined into one framework document, rather than left as three separate outputs that happen to relate to each other. The framework states which interfaces are covered, which controls close which weakness class, and what the measured trade-off is against the compliance-only baseline.

Validation: the primary validation is the quantitative result itself, whether the hardened design measurably reduces information leakage, control-plane CPU, and packet loss relative to baseline under the same attacks. Alongside that, I plan to walk the framework past two or three practitioners from my own professional network in telecom infrastructure and security, not as a formal expert panel, but as a

sanity check on whether the design assumptions (namespace-scoped RBAC, policy-engine admission control, Kata and node slicing together) hold up against how real multitenant clusters are actually run. Evaluation criteria: does the hardened configuration show a measurable, repeatable reduction on Wang et al.'s three metrics, and does the resource cost stay within a range a practitioner would consider reasonable for the density gained.

3.5 Ethical Considerations

All testing stays inside a lab-scale, non-production testbed built specifically for this research. No production telco infrastructure, live customer traffic, or third-party systems are targeted at any point. The attacks replayed in Phase 3 are limited to the four strategies Wang et al. (2025) already published and disclosed responsibly to the affected vendors; nothing new or undisclosed is being developed here. No personal or customer data is processed at any stage, and any practitioner input in Phase 4 is informal feedback on the design, not a data-collection exercise requiring separate consent procedures.

3.6 Limitations

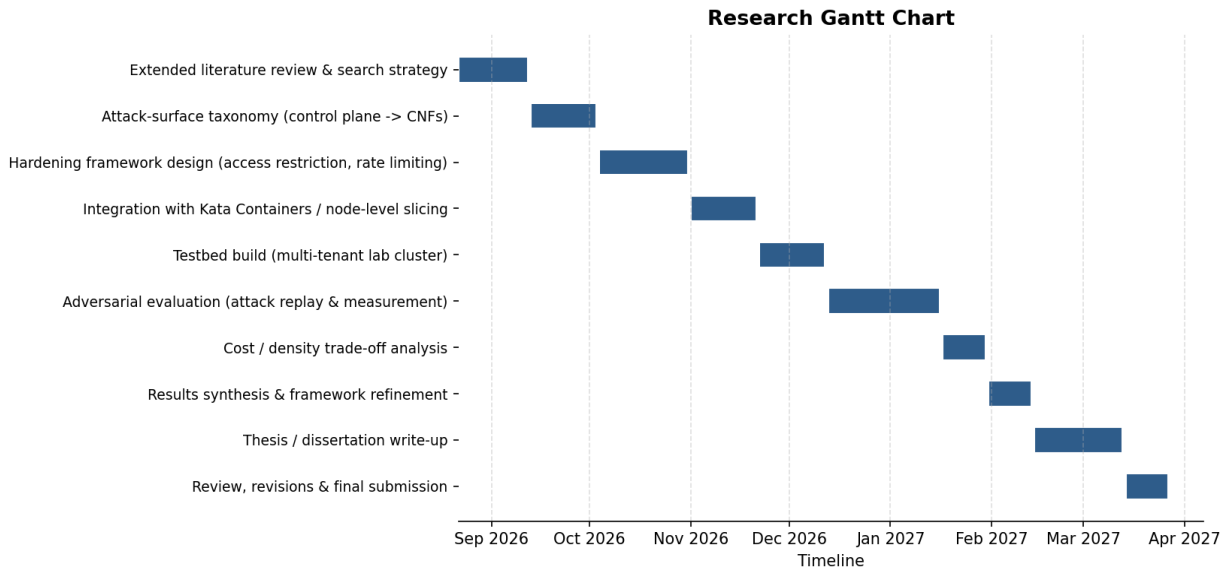
which is nowhere near production telco hardware, so absolute CPU and packet-loss numbers from Phase 3 will not transfer directly to a real operator's cluster. The relative comparison between hardened and unhardened configurations is the part expected to generalize, not the raw figures. Emulating 5G core network functions with free5GC or Open5GS instead of commercial AMF/SMF/UPF software is a second constraint of the same kind: it is the closest realistic substitute available without vendor access, but it is a substitute. The evaluation also only covers the four attack strategies Wang et al. (2025) documented, so it does not claim to cover every way a Kubernetes control plane can be attacked, and the research gap identified in Chapter 1 should be read as narrowed by this work, not closed by it. Finally, the course timeline, shown in the Gantt chart below, leaves limited room to repeat Phase 3 if the first testbed configuration turns out to be unstable, so some buffer is built into that phase rather than into the schedule as a whole.

3.7 Expected Outcomes

- An attack-surface taxonomy mapping Kubernetes control-plane interfaces, by weakness class, to the containerized 3GPP functions they can affect.
- A control-plane hardening design combining per-tenant access restriction and rate limiting with an existing compliance baseline.
- A description of how that design operates alongside Kata Containers and node-level slicing on one shared cluster.
- Evaluation results (information leakage, control-plane CPU, co-tenant packet loss, each averaged over repeated runs) comparing the hardened design against a compliance-only baseline, under the same attack strategies Wang et al. (2025) used.
- A resource-cost comparison intended to give a rough, evidence-based sense of what the hardening costs against the density it is meant to protect.

3.8 Gantt Chart

The schedule below assumes work starts at proposal submission (22 August 2026) and runs through evaluation and write-up. The testbed and attack-replay work carries the longest duration in the schedule because instability there is the most likely source of delay, and I would rather build that buffer in now than discover it is needed halfway through.



Milestone	Start	End	Duration
Extended literature review & search strategy	22 Aug 2026	12 Sep 2026	3 weeks
Attack-surface taxonomy (control plane -> CNFs)	13 Sep 2026	3 Oct 2026	3 weeks
Hardening framework design	4 Oct 2026	31 Oct 2026	4 weeks
Integration with Kata Containers / node-level slicing	1 Nov 2026	21 Nov 2026	3 weeks
Testbed build (multi-tenant lab cluster)	22 Nov 2026	12 Dec 2026	3 weeks
Adversarial evaluation (attack replay & measurement)	13 Dec 2026	16 Jan 2027	5 weeks
Cost / density trade-off analysis	17 Jan 2027	30 Jan 2027	2 weeks
Results synthesis & framework refinement	31 Jan 2027	13 Feb 2027	2 weeks
Thesis / dissertation write-up	14 Feb 2027	13 Mar 2027	4 weeks
Review, revisions & final submission	14 Mar 2027	27 Mar 2027	2 weeks

References

- Aiello, S. T., Rimal, B. P., Sheldon, F. T., & Wang, Y. (2026). Investigating security vulnerabilities in 5G control and user planes: Attack patterns and protection strategies. *Journal of Cybersecurity and Privacy*, 6(1), 37. <https://doi.org/10.3390/jcp6010037>
- Alboqmi, R., & Gamble, R. F. (2025). Enhancing microservice security through vulnerability-driven trust in the service mesh architecture. *Sensors*, 25(3), 914. <https://doi.org/10.3390/s25030914>
- Alnaim, A. K., Alwakeel, A. M., & Fernandez, E. B. (2022). Towards a security reference architecture for NFV. *Sensors*, 22(10), 3750. <https://doi.org/10.3390/s22103750>
- El-Hajj, M. (2025). Secure and trustworthy Open Radio Access Network (O-RAN) optimization: A zero-trust and federated learning framework for 6G networks. *Future Internet*, 17(6), 233. <https://doi.org/10.3390/fi17060233>
- Liu, P., Ji, S., Fu, L., Lu, K., Zhang, X., Lee, W.-H., Lu, T., Chen, W., & Beyah, R. (2020). Understanding the security risks of Docker Hub. In L. Chen, N. Li, K. Liang, & S. Schneider (Eds.), *Computer Security – ESORICS 2020* (LNCS vol. 12308, pp. 257–276). Springer. https://doi.org/10.1007/978-3-030-58951-6_13
- Morić, Z., Dakić, V., & Čavala, T. (2025). Security hardening and compliance assessment of Kubernetes control plane and workloads. *Journal of Cybersecurity and Privacy*, 5(2), 30. <https://doi.org/10.3390/jcp5020030>
- Pattaranantakul, M., He, R., Song, Q., Zhang, Z., & Meddahi, A. (2018). NFV security survey: From use case driven threat analysis to state-of-the-art countermeasures. *IEEE Communications Surveys & Tutorials*, 20(4), 3330–3368.
- Wang, C., et al. (2025). Losing control: Exposing security weaknesses of Kubernetes control plane interfaces. *Computers & Security*, 159, Article 103931.
- Şenel, B. C., Mouchet, M., Cappos, J., Fourmaux, O., Friedman, T., & McGeer, R. (2023). Multitenant containers as a service (CaaS) for clouds and edge clouds. *IEEE Access*, 11, 144574–144601.