Master Thesis
Connected Defence: Next-Generation Data Platform for Military Intelligence and Operations

First Lieutenant, Representative, Valentin Pfeil

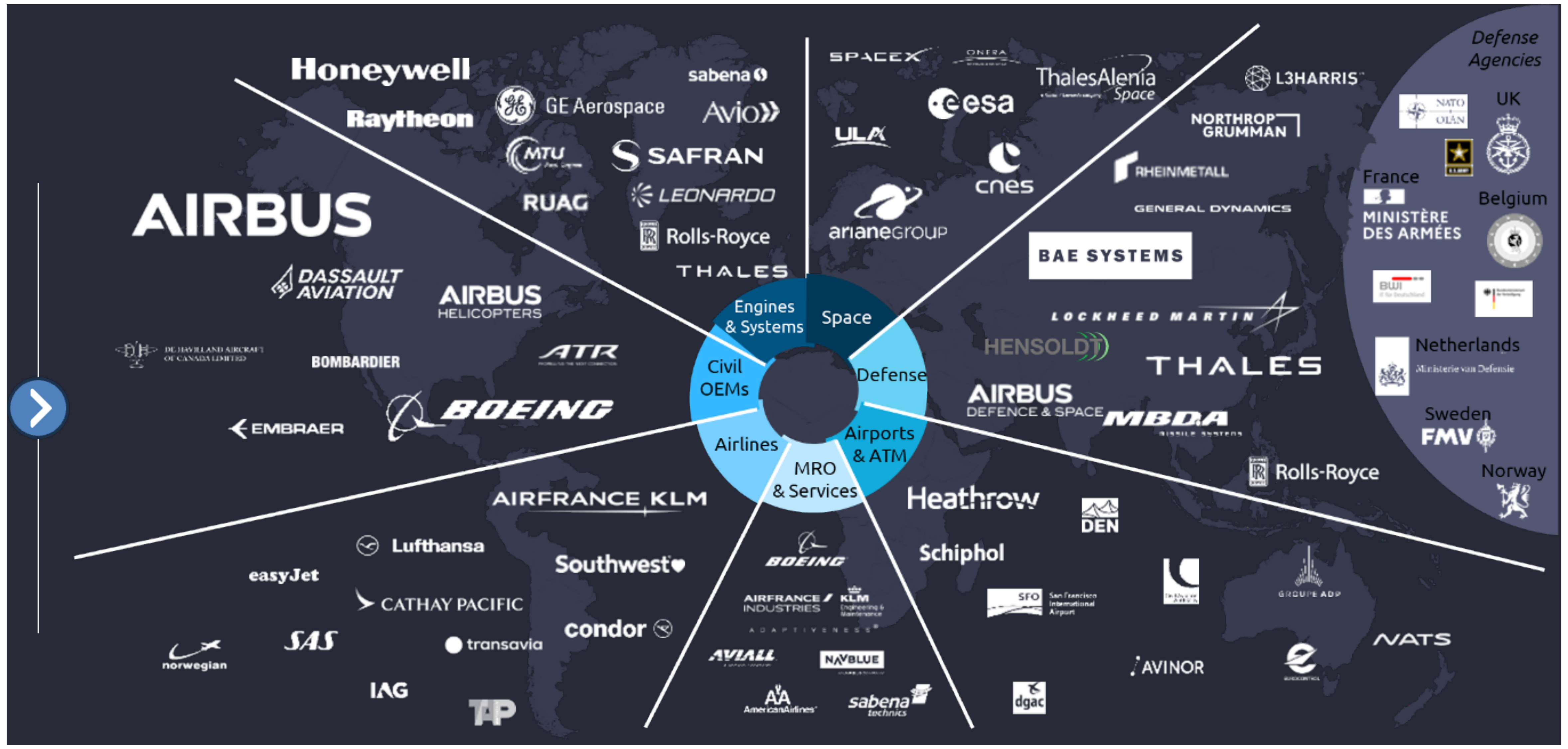
Capgemini

Stakeholder

“We operate in a world where technology is omni-present, profoundly transforming society, businesses and organizations. […]”

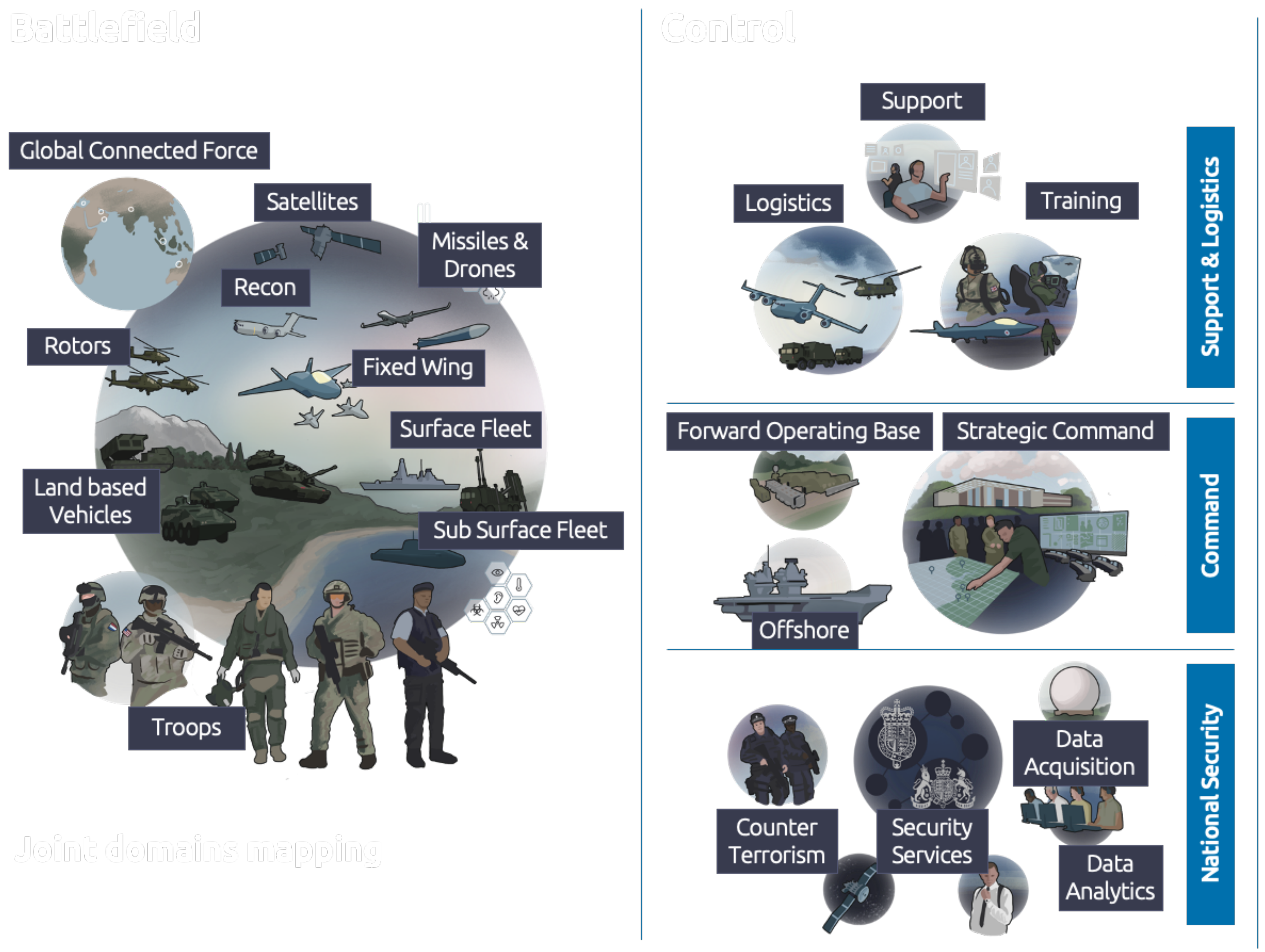
- Capgemini, CEO, Aiman Ezzat [2]

Strategic Collaboration Agreement (SCA) [3]
Connected Defence



Dislocation - Project OmniAware

- **Cloud-Native:** Leverages the scalability and flexibility of the AWS cloud infrastructure
- **Data-Driven:** Enables informed decision-making through real-time data analysis. (Telemetry, Images, etc.)
- **Defence-Compliant:** Meets stringent security and regulatory requirements. (Security Controls, Confidential Computing)
- **Monitoring and AI-Assisted Decision-Making:** Delivers continuous mission awareness through RT anomaly detection, model-driven threat assessments and adaptive alerting. Leverages the latest AI models with support for sensor fusion and dynamic retraining.



Research Questions

RQ1:

How can a cloud-native defence architecture be designed to ensure compliance with the **NATO Architecture Framework Version 4** (NAFv4) while supporting secure and scalable mission-critical operations?

RQ2:

What are the **key security challenges in defence cloud infrastructures** and how can a **confidential computing-based security model** be validated to ensure **compliance with defence security standards?**

RQ3:

How can **interoperability** between **cloud, edge** and **HPC** environments be ensured in a **defence cloud infrastructure** while maintaining **security** and **operational efficiency?**

Architecture, Deployment and Methodology

Architecture and Design (NAFv4/WAF)

- NATO Architecture Capability Team, ArchiMate Modeling Guide for NAFv4
- NAFv4, ArchiMate, Archi

Deployment Context and Implementation

- AWS
  - Accounts: GroupIT, AWS Guild Germany
  - Region: eu-west-1 (Ireland), eu-central-1 (Frankfurt)
- AWS CLI, CloudFormation, YAML-Templates, JSON Formats, Shell-/Python-Scripts

Architectural and Experimental Methodology

RQ1: Architectural modelling using NAFv4 conceptual views (e.g. NCV-2, NSOV-3, NSV-6) to derive compliance- and mission-driven system architecture.

RQ2: Implementation of Confidential Computing with at least two TEE nodes (Nitro Enclaves, AMD SEV-SNP) including Remote Attestation and Policy-based Secret Management via Vault with Logging.

RQ3: Development of secure interface layer (API Gateway, NGVA schema) to demonstrate interoperability and NATO compliance.

RQ1: How can a cloud-native defence architecture be designed to ensure compliance with the **NATO Architecture Framework Version 4** (NAFv4) while supporting secure and scalable mission-critical operations?

"It has resulted in the minimum number of ArchiMate element use to fulfil the needs of NAFv4, although there is some repetition of *object* usage. It is **not** intended to be a 1:1 mapping of ArchiMate to NAFv4." [4]

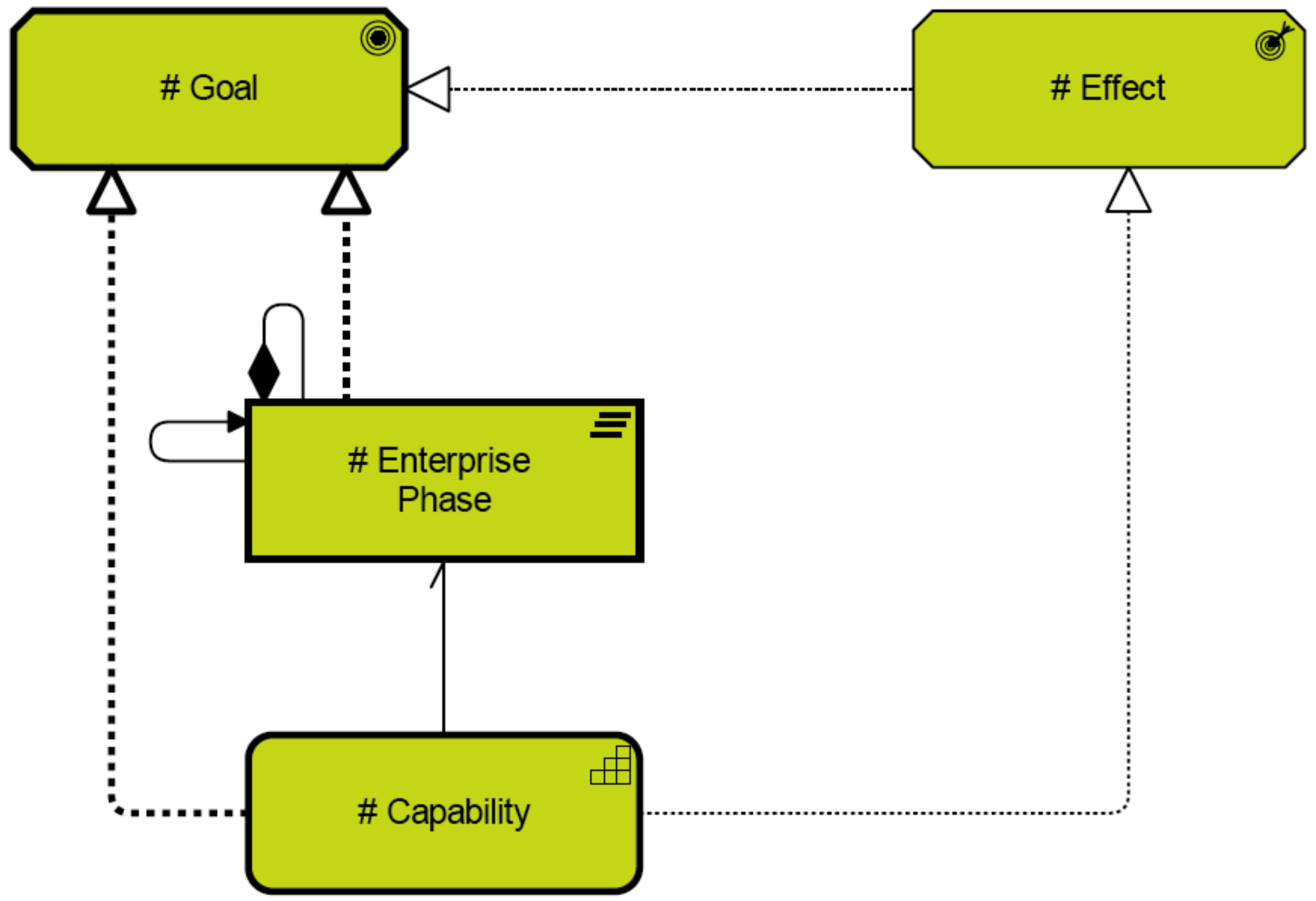


Figure 1: Example - NCV-1 [4]

Figure 2: NAF Grid [4]

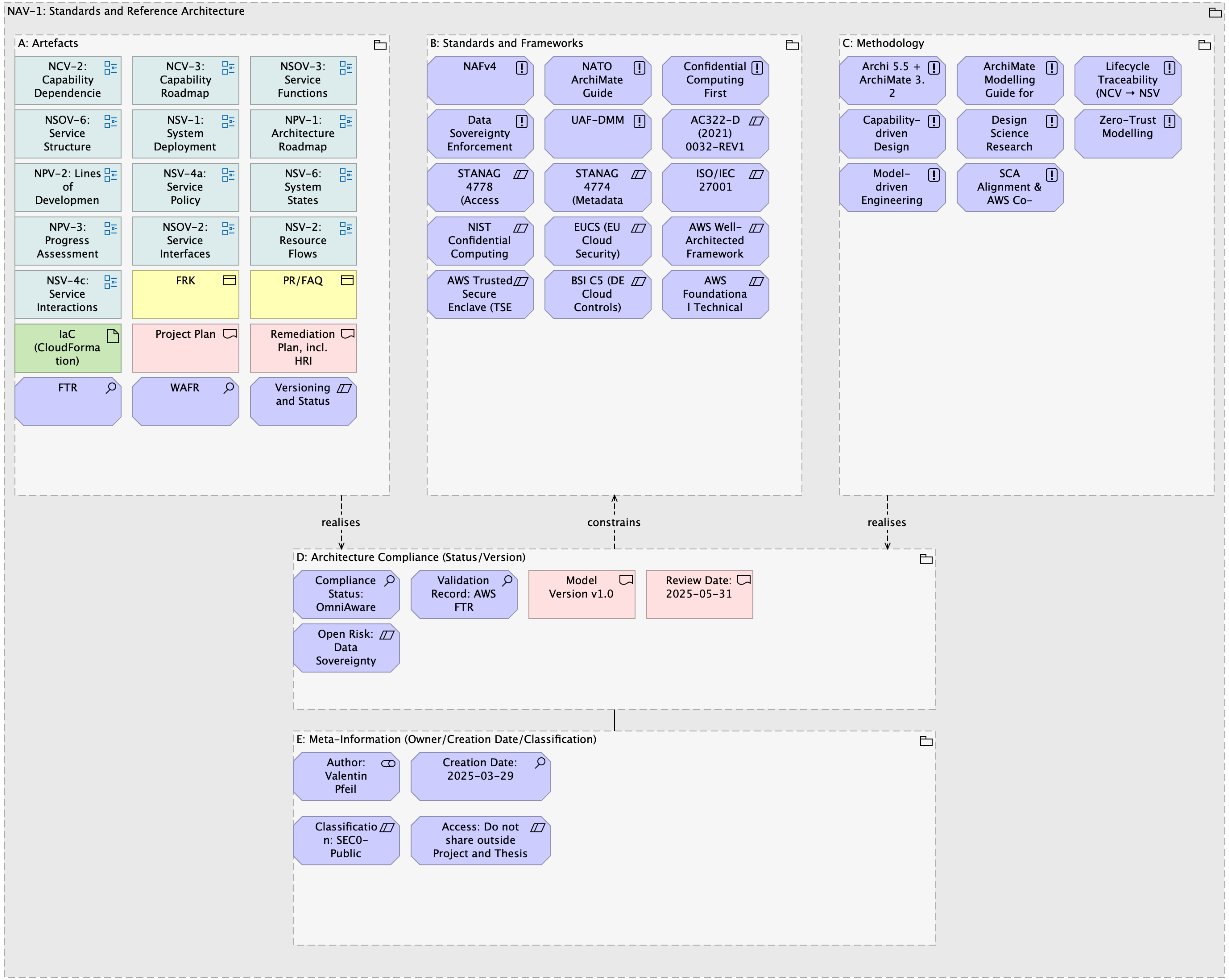


Figure 3: NAV-1 - OmniAware Standards and Reference Architecture

RQ1: How can a cloud-native defence architecture be designed to ensure compliance with the **NATO Architecture Framework Version 4** (NAFv4) while supporting secure and scalable mission-critical operations?

"AWS Well-Architected [...] Built around six pillars - operational, excellence, security, reliability, performance efficiency, cost optimization, and sustainability - [...] to evaluate **architectures** and implement scalable designs" [5]

Best Practices

- Drawing and diagramming tools: **Draw.io**, Creately, Figma, [...]
- AWS architecture icons

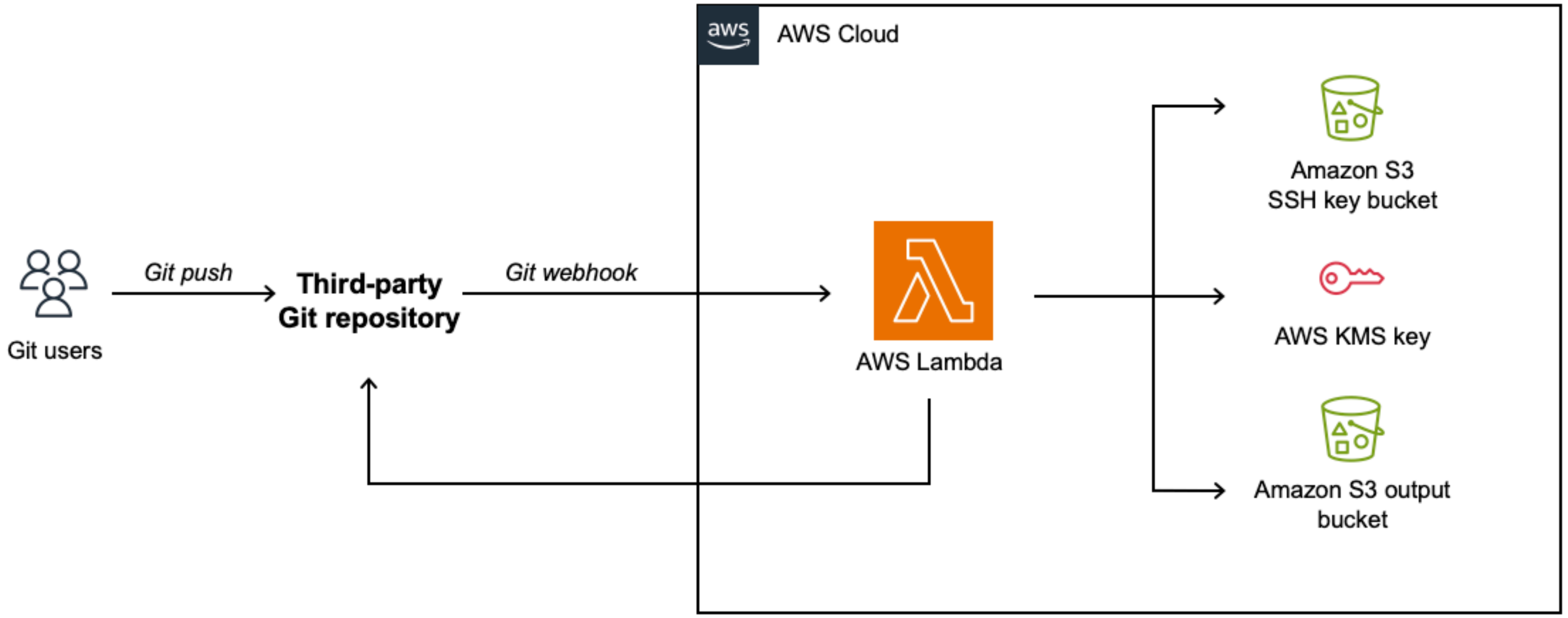


Figure 4: Example - Git to S3 Webhooks [5]

RQ1: How can a cloud-native defence architecture be designed to ensure compliance with the **NATO Architecture Framework Version 4** (NAFv4) while supporting secure and scalable mission-critical operations?

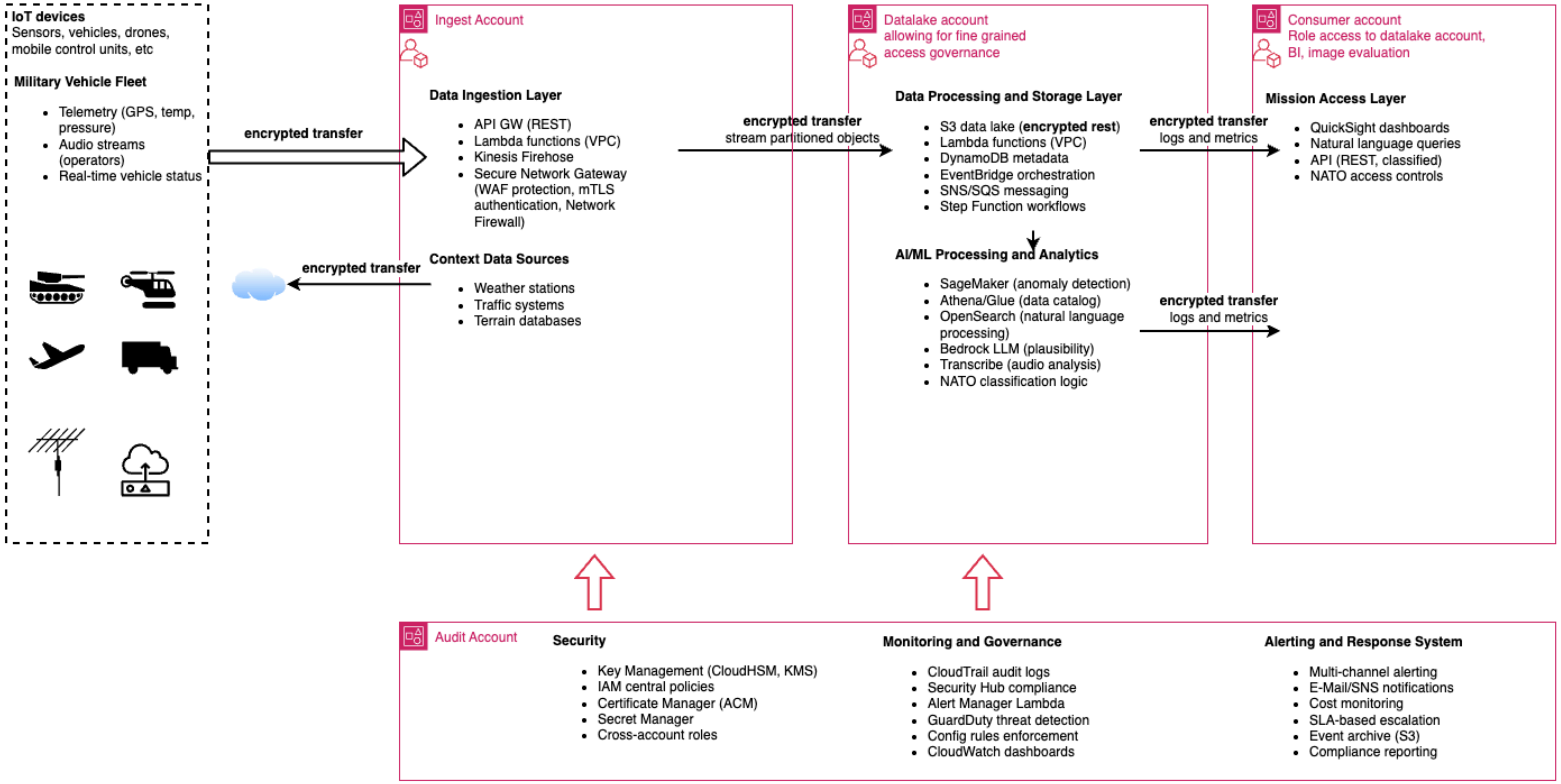


Figure 5: PHM - High-Level Overview of the Reference Architecture

RQ1: How can a cloud-native defence architecture be designed to ensure compliance with the **NATO Architecture Framework Version 4** (NAFv4) while supporting secure and scalable mission-critical operations?

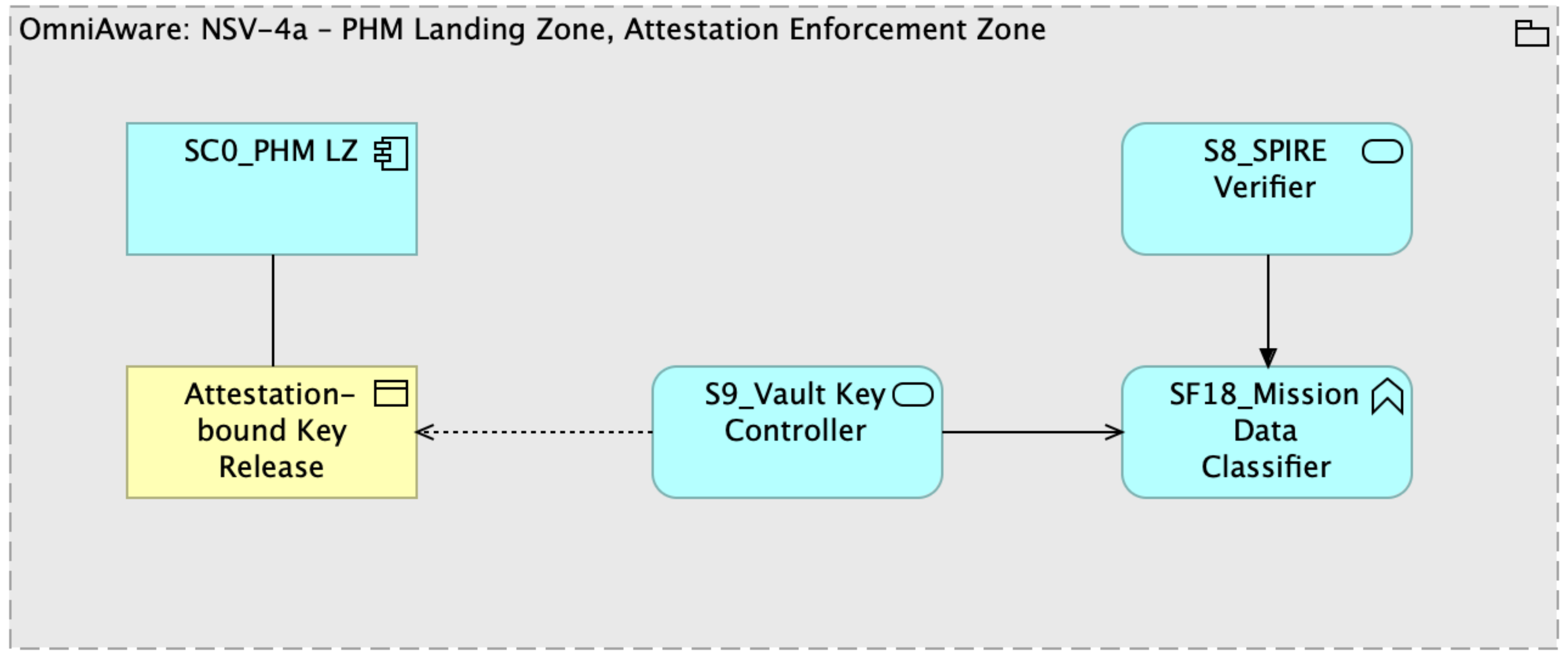


Figure 6: NSV-4a - PHM LZ Policy Enforcement

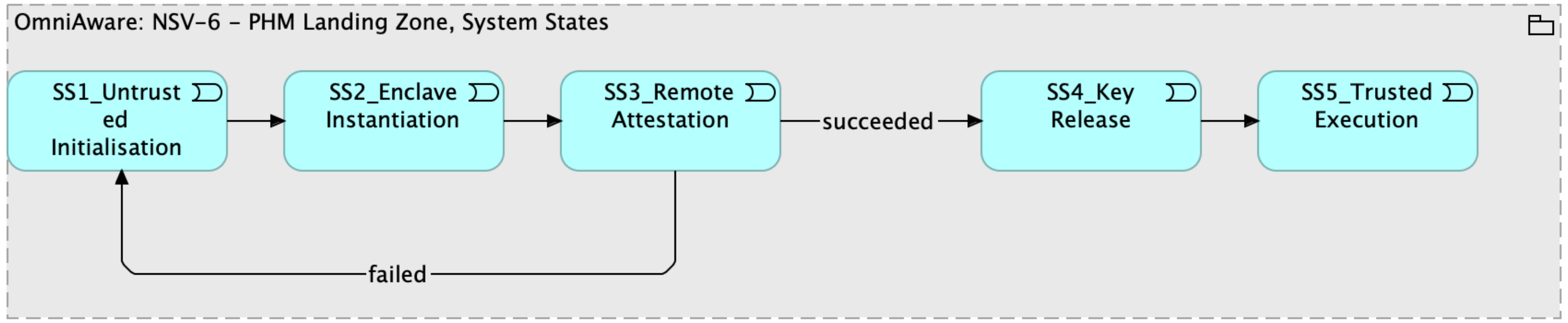


Figure 7: NSV-6 - PHM LZ System State Lifecycle

RQ2: What are the **key security challenges in defence cloud infrastructures** and how can a **confidential computing-based security model** be validated to ensure **compliance with defence security standards**?

Implementation

- AWS
  - Accounts: GroupIT, AWS Guild Germany
  - Region: eu-west-1 (Ireland), eu-central-1 (Frankfurt)

Standards, Frameworks and Best Practices

- AWS Well-Architected Framework
- AWS Foundational Technical Review

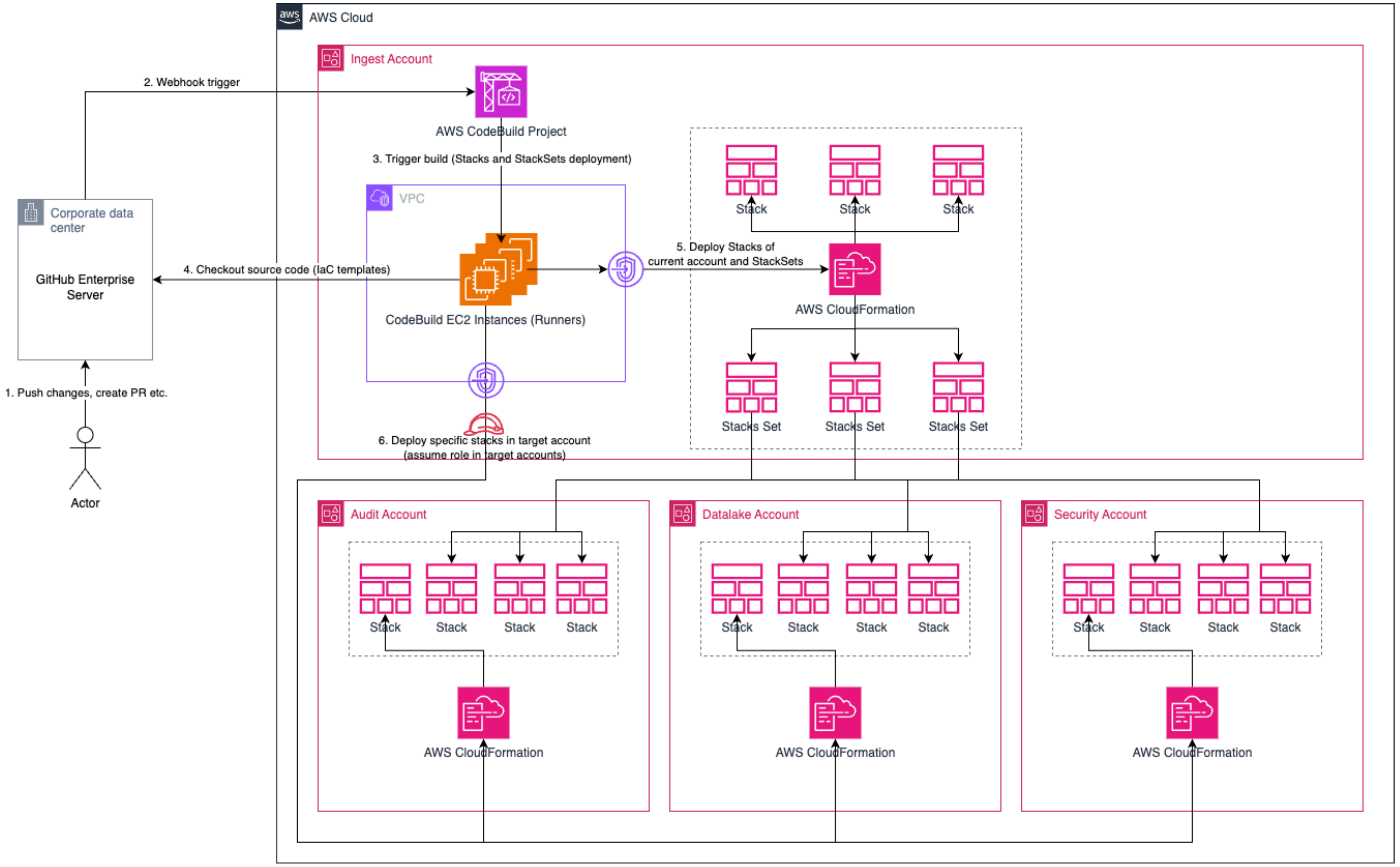


Figure 8: CI/CD Pipeline for Secure Deployment of Landing Zone Components

RQ2: What are the **key security challenges in defence cloud infrastructures** and how can a **confidential computing-based security model** be validated to ensure **compliance with defence security standards**?

Deployment Methodology for the Prototype

- Path A: Nitro Enclave-Based Remote Attestation
- **Path B: SEV-SNP-Based Remote Attestion**

Path B (completed) contained

- Environment Setup, Attestation Channel Setup, Vault Deployment and Joint Configuration
- Key Policy Enforcement, Test Secret Provision and Access, Validation and Logging
- Fully automated/partially automated deployment

Component Layer	Role in Attestation Workflow	Remarks
Confidential Runtime Environment	Hosts the trusted workload within a hardware-rooted enclave	**1x EC2 instance with SEV-SNP** **1x EC2 instance with Nitro Enclave-enabled**
`HashiCorp Vault (OSS)`	Key management service that enforces attestation-gated secret release	Deployed with TLS; runs standalone (or in dev mode for PoC)
Verifier Component	Validates attestation evidence against expected measurements and metadata	Implemented via `Vault` plugin or external policy enforcement module
Attestation Evidence Generator	Produces signed reports reflecting enclave state and identity	`sev-tool` (SEV-SNP) or Nitro Enclave SDK attestation interface
Secrets Policy Engine	Applies constraints for key release (e.g. PCR hash, enclave measurement, expiry)	Implemented via `Vault` HCL policy or custom validation logic
TLS Certificate Infrastructure	Secures communication between `Vault` and clients/verifiers	Self-signed or CA-issued; configured for `Vault` API endpoints
Test Secret (AES-256 key)	Validates the complete attestation-driven release workflow	Rotated regularly, used for decrypting synthetic mission payload

Table 1: Remote Attestation and Key Management Prototype

RQ2: What are the **key security challenges in defence cloud infrastructures** and how can a **confidential computing-based security model** be validated to ensure **compliance with defence security standards**?

```yaml
Resources:
  VaultInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: !FindInMap [RegionMap, !Ref "AWS::Region", UbuntuAMI]
      KeyName:
        !ImportValue
          Fn::Sub: "${InfraStackName}-KeyPair-Name"
      SubnetId:
        !ImportValue
          Fn::Sub: "${InfraStackName}-PrivateSubnet-ID"
      SecurityGroupIds:
        - !ImportValue
            Fn::Sub: "${InfraStackName}-Internal-Security-Group-ID"
      IamInstanceProfile:
        !ImportValue
          Fn::Sub: "${InfraStackName}-InstanceProfile-Name"
      UserData:
        Fn::Base64: !Sub |
          #!/bin/bash
          set -e

          hostnamectl set-hostname OmniAware-EC2-Vault
          echo '127.0.0.1 OmniAware-EC2-Vault' >> /etc/hosts

          snap install aws-cli --classic
          apt-get update && apt-get install -y jq curl wget git cmake
          ↪  build-essential \
            linux-headers-$(uname -r) libssl-dev pkg-config autoconf automake
            ↪  libtool \
            protobuf-compiler libprotobuf-dev gnupg
            ↪  software-properties-common unzip
```

Figure 9: Excerpt of Instance Deployment - OmniAware-EC2-Vault

```python
import jwt
from datetime import datetime, timedelta, timezone

private_key = open("private.key", "r").read()
payload = {
    "sub": "attester-001",
    "aud": "vault",
    "iss": "sev-snp",
    "nonce": "abc123",
    "iat": datetime.now(timezone.utc),
    "exp": datetime.now(timezone.utc) + timedelta(minutes=5),
    "report": open("/tmp/guest_report.b64", "rb").read().hex()
}
token = jwt.encode(payload, private_key, algorithm="RS256")
print(token)
```

Figure 10: Minimal Python tool to generate a signed SEV-SNP attestation JWT

RQ3: How can **interoperability** between **cloud, edge** and **HPC** environments be ensured in a **defence cloud infrastructure** while maintaining **security** and **operational efficiency**?

```json
{
    "id": "http://json-schema.org/draft-04/schema#",
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Core schema meta-schema",
    "definitions": {
        "schemaArray": {
            "type": "array",
            "minItems": 1,
            "items": { "$ref": "#" }
        },
        "positiveInteger": {
            "type": "integer",
            "minimum": 0
        },
        "positiveIntegerDefault0": {
            "allOf": [ { "$ref": "#/definitions/positiveInteger" }, { "default": 0 } ]
        },
        "simpleTypes": {
            "enum": [ "array", "boolean", "integer", "null", "number", "object",
            ↪ "string" ]
        },
        "stringArray": {
            "type": "array",
            "items": { "type": "string" },
            "minItems": 1,
```

Figure 11: Excerpt of JSON Schema Draft-04 - Sample Telemetry Schema for Test Purposes

```json
{
  "DateTime": {},
  "Vehicle_Configuration": {
    "Actual_Configured_Vehicle": {
      "vehicleId": {}
    }
  }
}
```

Figure 12: NGVA - Sample JSON Data Model, simplified

Path B - SEV-SNP

```
root@OmniAware-EC2-SEV-SNP-Ubuntu:/var/snap/amazon-ssm-agent/11320# sevctl ok
[ PASS ] - AMD CPU
[ PASS ]   - Microcode support
[ FAIL ]   - Secure Memory Encryption (SME)
[ PASS ]   - Secure Encrypted Virtualization (SEV)
[ FAIL ]     - Encrypted State (SEV-ES)
[ FAIL ]     - Secure Nested Paging (SEV-SNP)
[ SKIP ]       - VM Permission Levels
[ SKIP ]         - Number of VMPLs
[ PASS ]     - Physical address bit reduction: 0
[ PASS ]     - C-bit location: 51
[ PASS ]     - Number of encrypted guests supported simultaneously: 0
[ PASS ]     - Minimum ASID value for SEV-enabled, SEV-ES disabled guest: 0
[ FAIL ]     - SEV enabled in KVM: Error - /sys/module/kvm_amd/parameters/sev does not exist
[ FAIL ]     - SEV-ES enabled in KVM: Error - /sys/module/kvm_amd/parameters/sev_es does not exist
[ FAIL ]     - Reading /dev/sev: /dev/sev not readable: No such file or directory (os error 2)
[ FAIL ]     - Writing /dev/sev: /dev/sev not writable: No such file or directory (os error 2)
[ PASS ]   - Page flush MSR: DISABLED
[ FAIL ] - KVM supported: Error reading /dev/kvm: (No such file or directory (os error 2))
[ PASS ] - Memlock resource limit: Soft: 468017152 | Hard: 468017152
```

Figures 13: OmniAware-EC2-SEV-SNP - Excerpts of SEV-SNP status checks

```
root@OmniAware-EC2-SEV-SNP:/usr/bin$ dmesg | grep -i sev
[    0.652292] Memory Encryption Features active: AMD SEV SEV-ES SEV-SNP
[    0.880178] SEV: Using SNP CPUID table, 64 entries present.
[    1.411038] SEV: SNP guest platform device initialized.
[    6.173181] systemd[1]: Hostname set to <OmniAware-EC2-SEV-SNP>.
[    8.920888] sev-guest sev-guest: Initialized SEV guest driver (using vmpck_id 0)
```

Figures 14: OmniAware-EC2-SEV-SNP - Excerpts of SEV-SNP guest driver init

```
Attestation Report:

Version:                        4

Guest SVN:                      0

Guest Policy (0x30000):
  ABI Major:     0
  ABI Minor:     0
  SMT Allowed:   true
  Migrate MA:    false
  Debug Allowed: false
  Single Socket: false

Family ID:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Image ID:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

VMPL:                           1

Signature Algorithm:            1

Current TCB:

TCB Version:
  Microcode:   220
  SNP:         25
  TEE:         0
  Boot Loader: 4
  FMC:         None

Platform Info (39):
  SMT Enabled:                true
  TSME Enabled:               true
  ECC Enabled:                true
  RAPL Disabled:              false
  Ciphertext Hiding Enabled:  false
  Alias Check Complete:       true

Key Information:
    author key enabled: false
    mask chip key:      false
```

Figures 15: OmniAware-EC2-SEV-SNP - Excerpts of TCB attestation (1/2)

```
aws ssm start-session --target i-0d8d0ed8caba42511 --region eu-west-1

Committed TCB:

TCB Version:
  Microcode:   220
  SNP:         24
  TEE:         0
  Boot Loader: 4
  FMC:         None

Current Version:             1.55.31

Committed Version:           1.55.29

Launch TCB:

TCB Version:
  Microcode:   220
  SNP:         24
  TEE:         0
  Boot Loader: 4
  FMC:         None

Signature:
  R:
70 2B 3A 19 9B 89 1B 1C AE 32 63 B9 34 50 DE DF
27 0B 62 0A 7B ED 56 60 21 DA CE 6C 6D 8E 42 36
8A D9 6E 33 4C 48 C8 79 E9 12 1E D1 C2 3E 29 C2
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
  S:
06 C1 20 9D D2 81 A6 A1 71 86 E0 48 90 60 34 22
CA 9E 87 3B AA 91 27 37 C6 85 24 4C 55 EE 0D 41
74 C8 12 AB BE 33 CD A0 2F 27 A7 5F BD EF 03 52
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
root@OmniAware-EC2-SEV-SNP-Ubuntu:/var/snap/amazon-ssm-agent/11320#
```

Figures 16: OmniAware-EC2-SEV-SNP - Excerpts of TCB attestation (2/2)

```
root@OmniAware-EC2-Vault:/var/snap/amazon-ssm-agent/11320#        vault write -f transit/keys/attestation-test
Key                          Value
---                          -----
allow_plaintext_backup       false
auto_rotate_period           0s
deletion_allowed             false
derived                      false
exportable                   false
imported_key                 false
keys                         map[1:1749907186]
latest_version               1
min_available_version        0
min_decryption_version       1
min_encryption_version       0
name                         attestation-test
supports_decryption          true
supports_derivation          true
supports_encryption          true
supports_signing             false
type                         aes256-gcm96
```

Figure 17: OmniAware-EC2-Vault - Vault Key Attestation-Test - Transit Key Creation

```
root@OmniAware-EC2-Vault:/var/snap/amazon-ssm-agent/11320# vault write transit/encrypt/attestation-test plaintext=
$(echo -n "Hallo OmniAware" | base64)
Key             Value
---             -----
ciphertext      vault:v1:gL5CNusf80cRQf27GA8ti7suQNUT1XeuEj9U3JYbcQ6w3vd05zLD9YAk5Q==
key_version     1
```

Figure 18: OmniAware-EC2-Vault - Vault Message Encryption - Encoding of Plaintext with base64 and encryption with Transit Key

```
root@OmniAware-EC2-Vault:/var/snap/amazon-ssm-agent/11320# vault write transit/decrypt/attestation-test ciphertext
="vault:v1:gL5CNusf80cRQf27GA8ti7suQNUT1XeuEj9U3JYbcQ6w3vd05zLD9YAk5Q=="
Key            Value
---            -----
plaintext      SGFsbG8gT21uaUF3YXJl
```

Figure 19: OmniAware-EC2-Vault - Vault Message Decryption - Decryption with Transit Key

Path B - Vault, Secret Transit Engine

```
curl -sk --request POST \
  --url "$VAULT_ADDR/v1/auth/jwt/login" \
  --header "Content-Type: application/json" \
  --data "{\"jwt\": \"$JWT_TOKEN\", \"role\": \"sev-snp-role\"}"
{"request_id":"cc9e88f4-0c54-91d3-fb5e-cd706a520b3a","lease_id":"","renewable":false,"lease_duration":0,"data":nul
l,"wrap_info":null,"warnings":null,"auth":{"client_token":"hvs.CAESICbrRNAyQfKG7W3suaC8sMgh0JkH62756xvv7YLEKDIHGh4
KHGh2cy51VldEVjhaS1NMYmVrdWVIVHM2VDE3YUs","accessor":"PSIb6VPgxUmbUmsYFQPjEn6U","policies":["attestation-policy","
default"],"token_policies":["attestation-policy","default"],"metadata":{"role":"sev-snp-role"},"lease_duration":36
00,"renewable":true,"entity_id":"b31ad396-663f-0ecd-1821-658d1f5beb89","token_type":"service","orphan":true,"mfa_r
equirement":null,"num_uses":0},"mount_type":""}
root@OmniAware-EC2-SEV-SNP-Ubuntu:/opt/snpguest-test#
```

Figure 20: OmniAware-EC2-SEV-SNP-Ubuntu - JWT-Login via Remote Attestation

```
root@OmniAware-EC2-SEV-SNP-Ubuntu:/opt/snpguest-test# vault write transit/decrypt/attestation-test ciphertext="vau
lt:v1:VW1/P4nqSUHRDEb1CjEmiVAwNS6KtjThRjlj82tzTxI+GFMZ"
Key          Value
---          -----
plaintext    U0dWc2JnPT0=
root@OmniAware-EC2-SEV-SNP-Ubuntu:/opt/snpguest-test#
```

Figure 21: OmniAware-EC2-SEV-SNP-Ubuntu - Vault Message Decryption with Transit Key

```
aws ssm start-session --target i-05e8ce429e30b0fee --region eu-west-1

root@OmniAware-EC2-Vault:/var/snap/amazon-ssm-agent/11320# cat /var/log/vault/audit.log | jq
{
  "request": {
    "id": "2bf169ae-d3e3-f571-f4f2-d2f3974f8b34",
    "namespace": {
      "id": "root"
    },
    "operation": "update",
    "path": "sys/audit/test"
  },
  "time": "2025-06-15T17:13:31.375772546Z",
  "type": "request"
}
{
  "auth": {
    "accessor": "hmac-sha256:c85b42170c62be63fd91e27229e98bbed8014ffb6d7a587428271d1e3669da78",
    "client_token": "hmac-sha256:af102540ff28304357d3e5b516e2b1aa1b1c6afdc511d264a5979c6d1317ca29",
    "display_name": "root",
    "policies": [
      "root"
    ],
    "policy_results": {
      "allowed": true,
      "granting_policies": [
        {
          "type": ""
        },
        {
          "name": "root",
          "namespace_id": "root",
          "type": "acl"
        }
      ]
    },
    "token_policies": [
      "root"
    ],
    "token_issue_time": "2025-06-15T16:19:42Z",
    "token_type": "service"
  },
```

Figure 22: OmniAware-EC2-Vault - Vault Audit-Log-Events (1/2)

Secure Ingest Gateway - Image

```
~/Downloads/4.3_Scure-Ingest-API (0.41s)
cat 4.3_Secure-Ingest-API_Sample\ Picture.jpg | base64 > 4.3_Secure-Ingest-API_Sample\ Picture.jpg.txt
```

Figure 23: Encoding of Sample Picture with base64

/9j/4AAQSkZJRgABAgAAAQABAAD/
2wBDAA0JCgsKCA0LCgsODg0PEyAVExISEyccHhcgLikxMC4pLSwzOko+MzZGNywtQFdBRkxOUlNSMj5aYVpQYEpRUk//
2wBDAQ4ODhMREyYVFSZPNS01T09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT09PT0//
wgARCAVGCWADASIAAhEBAxEB/8QAGgABAAMBAQEAAAAAAAAAAAAAAAECAwQFBv/EABcBAQEBAQAAAAAAAAAAAAAAAAABAgP/
2gAMAwEAAhADEAAAAfZFyAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABMTAAASIkAESIkISCAAJIlBMSCJIJBAAAmAkEJITBMBMJE
SITAABMJIABMJIAmBMAkAIAJAIkAAAAAAAAAIkACBJBIAESIkIkBBMSIJIAJIAAAAAAAAAAAAAAAIJAIJAAAAhJCQQJgEwBJBIhJBIgAAJAAAAAAAAA
AAAAAAAAAAIJAAAAAAAAAAAAAAAAAAAAmBMATAJIAlATAJAAIkAAAAAAAACJISITAJCBMSAETBMTAlBKAkAESCBIIAAAAJIlATBMSImBKJAESIkESAA
IkAAAABBIAAAABBIAAAAAAAAESIkESITAAAAJIAAAAAAAAAAAAAAAAAiYAAAAJgJgABJAAAJAAAAAiQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJISI
mBMJISEJISBBIAAAAAAAAAAAAAESAIkEAmJIkBBIAETBIIkAISAAAESESISESAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABJAAAAAAAAESESISISISIAAAAAAAA
AAAAAAAiRBJAAAAAAAAAAAJAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJIAAAAABKJETBIIkAAAAAAAAAAAAAAAAUAEAARIhIECYkAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAABRJAQAASsEkJEAJhBKxIRIIkImASQmAAAEAAAAARIgABMAAkgABMAAAACJgmAAAAmAAAAAAAAAkAAAAAAAAAAAAAAAAAAAAAkgAAAA
AAAkgAAABMBMAAkgkhMAkhMEoEoEokAAAAAAAAAIkBQAAAAAAQAAAAAAAAAABEgAAAmFJhAAAAAAAAABKxIRIAIkkSKhJEoExJCYJiYCRAEhEgAAAAAA
AAAAABCRCRCRAAAAAQCCSEwACSAJgAAAACABMAAAAAAAAAAAAACQAAAAAAAAAAAAAAAAAAATAACSAACSCSCSACSEiJgEiEiCSEwAAJACJiSJAAAAAAAAA
AAAAAFAAATEkAAAAABAAUEBQAQFBAUAAmAAAAAEATAAAEkJKAAAABEhEgiREgRIiQAiQAAAAATATAATAJITAAAAAABCQiYAAAAESRAAAAAACCSCYAAAA
AAAAAAAAACtgRIkAAkhIgAAAAAAAAEwAAAAAkiYExIhIiQARIAAAiQAARIAAiQiQiYkAAIkAAAAAAAAABQAQAAFAAAAAmJBEiAAAAAAAAABAUEBQAAAAAA
QmFTAmAAAABEigAAAAAAAAACCQAAAAACSEwJCEiJiSJABEhEwAAAAAAImAAAEEAAAAAAAACJgAAAAAAAAAAAAAAAAAlMExIhIAhIhMABMABMACYkiQgAk
gkhMBMCYCQAAAAAAAAAAAAAAAAAAAAAAAAAAABQQFAAJgBAAUQSAACUCQRNZAAAAAAEwIkAAQFAAAAAAABAUAAAEAAlEqAAAAAAAiYJARJEgAAAAAmBIAA
AAAAABAAAAAABAAAAREwAAAAAAAAARMAAAAAAAAAAAAAAAAACJiSwAAAAAAISIJIlAJImJCJCBMSIkAAAAAAAAAUAEBQQFBAAAAAAAAAAAAAAAAUAAAEBQAAC
CSgqYkAAAAAAAAAAAmAAAABAUAAEBQAAAQFBAAUmAmCQAAAAAAAAAAAAAASgJQJBEgQSAAAAAAAACEiAAEABE5pogsoIAAAAAAAARJBJAAAAAAAAAAAAAAA
AAAALBQQFBIkBBICJCJISCJAESIkBBIAAAUEABQAAAAAAAAAQFABAAUAEAAAAAABQAAQAFEJKAAACpgSAAAAAAAEBQAAAAAQAAAFAAAAABAUAAkCCUSA
AAAACCYCUSAAAAAAJgSAAAAiQAiQAAAQTAAIAAiURMAkgAAAAAAAAAACJgAAAAAAAAiQAAAAAAAAtEgAFBAAVAgCQIEokIkAARIAAABQACYAIkAAAAAA
AAQFAAAAAABAUAAEBQAAASJAQTEwAAAoIAAmAmAkAUiUAABQQCJAFBAAAUAAAAEBQAQFAAAAAlAAmEkAAAAAkAABEgACYEoEoEwBMCYEoEgEEoEoEwAA
gEEoJMAAAAAAARIAARIRIAAiQgAAAAAAAAAAAAAAAAAAJmJAUgSQkolQBCJQTEwEiEgAiQAAFiQAAAAAAAAAAAAAABAAAAUEBQACJQFAAAAABAUEBUSSAAAA
AAAAAAAAJgSiQFBAAAAUEAAABQAQAAAAAAAAAAAAFAAAAAEkSgSAEJglEgAAAAEwACYAAAAAAABEkARMCYAIAAAAAAAiQhJEoJRIICYAAAAAAAAAESAAAIk
ACJAESIlBYEoAAAEwEwkQAExMAEwCYEoEhQAAAAAAAAAAAAAAAAQFAABAAUABEkAABQQAAAAAgAAAAAAAAAARIAAACSJAFBAAUEAVsAAAAAAAAAAAAAAAAUAE
AABQAAJgAAEgABEgAAAAAAAAAAAAIJgABACAAAAAAAAAAAImCYkRMAkIAAASQAAAAAAAAABEhCSJAAQSQWAAAAAAAAAAJIAAABIUAiQAAAAAAAAAAAAEBQQFB

Figure 24: Excerpt of base64 encoded Sample Picture

Figure 25: Sample Picture

Introduction | Thesis | **Validation** | Results | Conclusion

Insights - Architecture, Design and Implementation

NAFv4-Driven Modelling Approach

- Strategic-to-runtime traceability via NSV, NPV and NSOV views
- Systematic decomposition aligned with mission and compliance needs

Security Architecture via Trusted Views

- NSV-4a/6 and NPV-3 modelled trust anchors, attestation flows and key usage
- Interfaces (NSOV-2/3) implemented as zero-trust API boundaries

TEE-Based Security Execution with Vault

- Dual-path attestation validated via AWS Nitro Enclaves and AMD SEV-SNP
- JWT-based trust workflows confirmed cryptographically and operationally
- Vault Transit Engine enforced data-in-use protection with policy-bound secrets

Ingestion Pipeline

- NGVA API Gateway and structured logs enabled secure ingest to AWS Datalake

Future Extensibility

- The design allows optional integration of sensor modules, Digital Twin simulations and real-time data visualisation layers
- Architecture supports modular extension without compromising core trust primitives
- Vault-based architecture and JWT workflows are modular and extendable to other policy engines or enclaves
- Prototype components (API Gateway, telemetry ingestion) can be hardened and scaled via IaC (e.g. Terraform, OPA)
- Next-gen extensions could target fully automated trust pipelines and policy-controlled data access

Challenges

01.02.2025 - 30.06.2025

Phase I: Limited project maturity and parallel exam preparation reduced available focus and continuity.

01.02.2025 - 31.03.2025

Phase II: Increased project scope and involvement in strategic and BD-related tasks led to competing priorities and fragmented capacity.

01.04.2025 - 31.05.2025

Phase III: Tight timelines and coordination efforts across stakeholders posed significant constraints on implementation and documentation.

01.06.2025 - 21.06.2025

Phase IV: Final synchronisations under deadline pressure, including printing logistics and latency, introduced additional stressors.

22.06.2025 - 30.06.2025

Cross-phase

- Balancing defence-grade implementation depth with academic formalism
- Aligning security design iterations with rapidly evolving AWS primitives
- Coordinating distributed team input across time zones and priorities
- Translating complex experimental architecture (e.g. Confidential Computing, Remote Attestation) into reproducible thesis artefacts
- Managing dual publication requirements (academic and industrial) without overlap or disclosure risk

Evaluation and Outlook

Research Answers

- Architecture is **NAFv4**-compliant, mapped to **NATO** models and implemented using formalised **cloud-native** views (via ArchiMate).
- Key security challenges such as **trust gaps** and **classified data** have been mitigated through **TEE**-based **policy enforcement**, Vault integration and **attestation**.
- A secure, **interoperable** interface architecture was implemented, separating **data/control** planes and aligning with **zero-trust networking** principles in line with **NATO** guidelines

OmniAware sets the stage for a trusted digital doctrine, enabling **sovereign**, **NATO**-aligned defence architectures that scale from **PoC** to full **operational readiness**. Its adaptable blueprint can inform future procurement, certification and capability planning initiatives across multi-domain coalitions.

Operational Integration

Platform design supports integration with sensor networks, mission systems and simulation tools.

Future Extensions

Next steps include RT visualisation, predictive simulations and AI-based decision support via Digital Twin and confidential analytics pipelines. Potential extensions include systems with TEE-based execution for tactical integrity and operational safety.

Scalability Across Domains

The system is modular and scalable across NATO, EU and national deployments, supporting both edge and HPC use cases

References

[1] Capgemini, 2024 Integrated Annual Report, Paris, France, annual report, May 7, 2025. Available: https://reports.capgemini.com/2024/en/.

[2] Capgemini, 2024. Capgemini and AWS expand strategic collaboration to enable broad enterprise Generative AI adoption. [Online]. Available: https://www.capgemini.com/news/press-releases/capgemini-and-aws-expand-strategic-collaboration-to-enable-broad-enterprise-generative-ai-adoption/

[3] NATO Architecture Capability Team, 2025. ArchiMate Modeling Guide For the NATO Architecture Framework Version 4. [Online]. Available: https://www.nato.int/nato_static_fl2014/assets/pdf/2025/2/pdf/2502-NAFv4-ArchiMate.pdf.

[4] Amazon Web Services, 2024. [Online]. Available: https://aws.amazon.com/architecture/well-architected/.

[5] J. Salvermoser and V. Pfeil, 2025. OmniAware Use Case - Reference Architecture, Reference Architecture, Capgemini Internal.

Capgemini

Valentin Pfeil
Institute for Software Technology
Research Institute CODE
University of the Bundeswehr Munich
valentin.pfeil@unibw.de
https://www.unibw.de/code

FI CODE Forschungsinstitut Cyber Defence Universität der Bundeswehr München

Q & A

Valentin Pfeil
Institute for Software Technology
Research Institute CODE
University of the Bundeswehr Munich
valentin.pfeil@unibw.de
https://www.unibw.de/code

Forschungsinstitut Cyber Defence
Universität der Bundeswehr München