System Design

Foundational Platform Service v2.0

Version: 1.0 **Author:** John Gasser **Status:** Draft for Review

Table of Contents

1. Reference Architecture
2. Multi-Cloud Abstraction
3. Zero Trust Controls
4. Compliance & ATO Acceleration
5. Continuous Monitoring
6. Cross-Domain & Edge
7. Threat Model (STRIDE Analysis)

1. Reference Architecture

Three-Plane Architecture

The platform is organized into three logical planes that separate concerns while maintaining cohesive security controls.

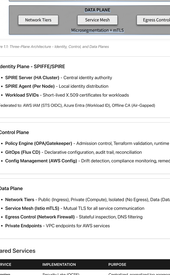

Figure 1.1: Three-Plane Architecture - Identity, Control, and Data Planes

Identity Plane - SPIFFE/SPIRE

- **SPIRE Server (HA Cluster)** - Central identity authority
- **SPIRE Agent (Per Node)** - Local identity distribution
- **Workload SVIDs** - Short-lived X.509 certificates for workloads

Federated to: AWS IAM (STS OIDC), Azure Entra (Workload ID), Offline CA (Air-Gapped)

Control Plane

- **Policy Engine (OPA/Gatekeeper)** - Admission control, Terraform validation, runtime enforcement
- **GitOps (Flux CD)** - Declarative configuration, audit trail, reconciliation
- **Config Management (AWS Config)** - Drift detection, compliance monitoring, remediation

Data Plane

- **Network Tiers** - Public (Ingress), Private (Compute), Isolated (No Egress), Data (Database)
- **Service Mesh (Istio mTLS)** - Mutual TLS for all service communication
- **Egress Control (Network Firewall)** - Stateful inspection, DNS filtering
- **Private Endpoints** - VPC endpoints for AWS services

Shared Services

SERVICE	IMPLEMENTATION	PURPOSE
Logging	Security Lake (OCSF)	Centralized, normalized log aggregation
SIEM Feed	Security Hub + Splunk	Alert correlation and incident response
KMS/HSM	CloudHSM-backed KMS	FIPS 140-3 Level 3 key management
Artifact Registry	ECR + Cosign	Signed container images with SBOM
Secrets	External Secrets Operator	Cross-cloud secrets abstraction

Network Tiers

Figure 1.2: VPC Network Architecture with subnet tiers across availability zones

Egress Control

All egress traffic flows through centralized inspection:

Figure 1.3: Egress Control Flow through multiple security layers

Artifact Supply Chain

Figure 1.4: Artifact Supply Chain with signing and admission control

2. Multi-Cloud Abstraction

Environment Contract Interface

The platform uses a standardized interface that abstracts cloud-specific differences:

```
# Core Configuration (identical across all clouds)
variable "environment_name"   { type = string }
variable "environment_type"   { type = string }  # production, staging, development
variable "classification_level" { type = string }  # U, S/C, S/A
variable "region"             { type = string }  # Mapped to cloud-specific region

# Required Tags (enforced by policy)
variable "required_tags" {
  type = object({
    Owner       = string
    Environment = string
    Classification = string
    CostCenter  = string
    Compliance  = string
  })
}

# Network Abstraction
variable "network_config" {
  type = object({
    vpc_cidr_block      = string
    availability_zone_count = number
    subnet_tiers = map(object({
      enabled = bool
      cidr_bits = number
    }))
  })
}

# Kubernetes Abstraction
variable "kubernetes_config" {
  type = object({
    cluster_name  = string
    cluster_version = string
    node_groups   = map(object({...}))
  })
}
```

Cloud Provider Mappings

ABSTRACTION	AWS GOVCLOUD	AZURE GOVERNMENT	AIR-GAPPED (K3S)
Virtual Network	VPC	VNet	Host Network / VXLAN
Subnets	Subnet	Subnet	Network Policies
Load Balancer	ALB/NLB	Azure LB	MetalLB
NAT	NAT Gateway	NAT Gateway	Iptables
Kubernetes	EKS	AKS	K3s
Node Groups	Managed Node Groups	Node Pools	Server Nodes
IAM	IAM Roles	Managed Identity	RBAC
Workload Identity	IRSA	Workload Identity	Service Accounts
Key Management	KMS + CloudHSM	Key Vault + HSM	LUKS + TPM
Private Endpoints	PrivateLink	Private Link	NJA (direct)
Logging	CloudWatch	Azure Monitor	Loki
Audit	CloudTrail	Activity Log	auditD/tac
Threat Detection	GuardDuty	Defender	Falco
Config Mgmt	AWS Config	Azure Policy	Custom

Multi-Cloud Deployment Architecture

Figure 2.1: Multi-Cloud Deployment showing AWS GovCloud and Azure Government implementations

AWS GovCloud Implementation

```
terraform/modules/aws-environment/
├── vpc.tf          # VPC, Subnets, NACLs, Route Tables, NAT, Flow Logs
├── security.tf     # Security Groups, NACLs
├── kms.tf          # KMS Keys, S3 for CloudTrail
├── eks.tf          # EKS Cluster, Node Groups, Add-ons
├── iam.tf          # IAM Roles, Policies, IRSA
├── cloudtrail.tf   # CloudTrail, S3 Bucket, KMS
├── logging.tf      # CloudWatch, CloudTrail, GuardDuty, Config
├── transit.tf      # Transit Gateway, VPN
├── variables.tf    # Environment Contract Interface
└── outputs.tf      # Standardized Outputs
```

Azure Government Stub

```
terraform/modules/azure-environment/
├── vnet.tf         # VNet, Subnets, NSGs, NAT, Flow Logs
├── aks.tf          # AKS Cluster, Node Pools, Key Vault
├── monitoring.tf   # Log Analytics, Azure Monitor
└── outputs.tf      # Same output structure as AWS
```

3. Zero Trust Controls

Strong Identity: SPIFFE/SPIRE

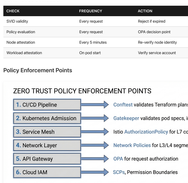

Figure 3.1: SPIFFE/SPIRE Identity Architecture with workload attestation

mTLS Everywhere

All service-to-service communication uses mutual TLS with SPIFFE SVIDs:

1. **Service Mesh (Istio Ambient)**: Ztunnel provides L4 mTLS
2. **Database Connections**: PostgreSQL with certificate authentication
3. **External Services**: AWS services via PrivateLink with IAM auth

Microsegmentation

```
# Kubernetes Network Policy - Default Deny
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny-all
  namespace: tenant-a
spec:
  podSelector: {}
  policyTypes:
  - Ingress
  - Egress

# Istio AuthorizationPolicy - Allow specific SPIFFE IDs
apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: frontend-to-backend
  namespace: tenant-a
spec:
  selector:
    matchLabels:
      app: backend
  action: ALLOW
  rules:
  - from:
    - source:
        principals:
        - "spiffe://platform.gov/ns/tenant-a/sa/frontend"
```

Continuous Evaluation

CHECK	FREQUENCY	ACTION
SVID validity	Every request	Reject if expired
Policy evaluation	Every request	OPA decision point
Node attestation	Every 5 minutes	Re-verify node identity
Workload attestation	On pod start	Verify service account

Policy Enforcement Points

4. Compliance & ATO Acceleration

NIST 800-53 Mapping Approach

Representative Controls Mapped:

FAMILY	CONTROL	PLATFORM IMPLEMENTATION	INHERITANCE
AC	AC-2 Account Management	Okta + Ansible RBAC	CSP: 30%, Platform: 50%, App: 20%
AC	AC-3 Access Enforcement	OPA policies	Platform: 80%, App: 20%
AC	AC-6 Least Privilege	IRSA + Permission Boundaries	Platform: 70%, App: 30%
AU	AU-2 Audit Events	CloudTrail + K8s audit	CSP: 30%, Platform: 50%
AU	AU-3 Audit Content	Structured logging	Platform: 100%
AU	AU-6 Audit Review	Security Hub dashboards	Platform: 80%, App: 20%
SC	SC-7 Boundary Protection	VPC + NACLs + Firewall	CSP: 30%, Platform: 70%
SC	SC-8 Transmission Confidentiality	mTLS everywhere	Platform: 100%
SC	SC-13 Cryptographic Protection	KMS + CloudHSM	CSP: 40%, Platform: 60%
CM	CM-2 Baseline Configuration	Terraform + GitOps	Platform: 100%
CM	CM-3 Configuration Change Control	PR-based GitOps	Platform: 100%
CM	CM-6 Configuration Settings	OPA policies	Platform: 80%, App: 10%
IA	IA-2 Identification	SPIFFE/SPIRE	Platform: 100%
IA	IA-5 Authenticator Management	Secrets Manager + rotation	Platform: 80%, App: 20%

Inheritance Model

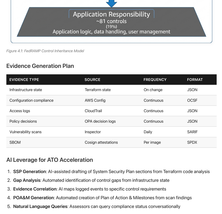

Figure 4.1: FedRAMP Control Inheritance Model

Evidence Generation Plan

EVIDENCE TYPE	SOURCE	FREQUENCY	FORMAT
Infrastructure state	Terraform state	On change	JSON
Configuration compliance	AWS Config	Continuous	OCSF
Access logs	CloudTrail	Continuous	JSON
Policy decisions	OPA decision logs	Continuous	JSON
Vulnerability scans	Inspector	Daily	SARIF
SBOM	Cosign attestations	Per image	SPDX

AI Leverage for ATO Acceleration

1. **SSP Generation**: AI-assisted drafting of System Security Plan sections from Terraform code analysis
2. **Gap Analysis**: Automated identification of control gaps from infrastructure state
3. **Evidence Correlation**: AI maps logged events to specific control requirements
4. **POA&M Generation**: Automated creation of Plan of Action & Milestones from scan findings
5. **Natural Language Queries**: Assessors can query compliance status conversationally

5. Continuous Monitoring

Compliance-as-Code

Figure 5.1: Continuous Compliance Pipeline from commit to monitoring

Runtime Checks

CHECK	TOOL	FREQUENCY	ALERT THRESHOLD
Image vulnerabilities	Inspector	On push + daily	Critical/High
Runtime behavior	Falco	Continuous	Any suspicious
Network anomalies	GuardDuty	Continuous	Medium+
Config compliance	AWS Config	15 minutes	Any non-compliant
Certificate expiry	cert-manager	Hourly	< 30 days

Drift Detection

```
# AWS Config Rule for drift detection
resource "aws_config_config_rule" "s3_encryption" {
  name = "s3-bucket-server-side-encryption-enabled"
  source {
    owner             = "AWS"
    source_identifier = "S3_BUCKET_SERVER_SIDE_ENCRYPTION_ENABLED"
  }
}

# Auto-remediation via SSM
resource "aws_config_remediation_configuration" "s3_encryption" {
  config_rule_name = aws_config_config_rule.s3_encryption.name
  target_id        = "AWS-EnableS3BucketEncryption"
  target_id        = "AWS-EnableS3BucketEncryption"
  automatic        = true
}
```

Dashboards

DASHBOARD	PURPOSE	UPDATE FREQUENCY
Security Hub	Aggregated security findings	Real-time
Compliance Scorecard	Control satisfaction percentage	Daily
Drift Monitor	Configuration changes	15 minutes
Cost Tracker	Budget vs. actual	Daily
SLA Monitor	TTE, availability metrics	Real-time

6. Cross-Domain & Edge

CDS Integration Pattern

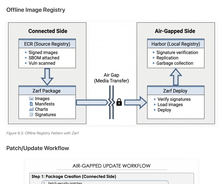

Figure 6.1: Cross-Domain Solution Integration Pattern

Abstract CDS Adapter Interface

```
// CDSAdapter defines the interface for cross-domain data transfer
type CDSAdapter interface {
    // PrepareExport prepares data for export to lower classification
    PrepareExport(ctx context.Context, data []byte,
        metadata ExportMetadata) (*ExportPackage, error)

    // ValidateImport validates and sanitizes imported data
    ValidateImport(ctx context.Context,
        pkg *ExportPackage) (*ValidatedData, error)

    // GetTransferStatus checks status of pending transfer
    GetTransferStatus(ctx context.Context,
        transferID string) (*TransferStatus, error)

    // RequestApproval initiates manual approval workflow
    RequestApproval(ctx context.Context, pkg *ExportPackage,
        approvers []string) (*ApprovalRequest, error)
}

type ExportMetadata struct {
    SourceClassification string
    TargetClassification string
    DataType             string
    Checksum             string
    Requestor            string
    Justification        string
}
```

Store-and-Forward for D/DIL

Figure 6.2: Store and Forward for Disconnected Operations

Offline Image Registry

Figure 6.3: Offline Registry Pattern with Zarf

Patch/Update Workflow

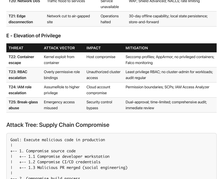

Figure 6.4: Air-Gapped Update Workflow

7. Threat Model (STRIDE Analysis)

Scope

This threat model covers the Foundational Platform Service with focus on:

- Identity and access management
- Supply chain integrity
- Network boundaries (egress/ingress)
- Edge/air-gapped deployment compromise

STRIDE Threat Analysis

S - Spoofing

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T1: Workload identity spoofing	Attacker compromises pod and attempts to assume another workload's identity	Unauthorized access to services	SPIFFE attestation requires valid K8s SA + node identity
T2: Node identity spoofing	Attacker spins up rogue node claiming to be legitimate	Malicious workloads join cluster	AWS Instance Identity Document validation, TPM attestation for edge
T3: User impersonation	Stolen credentials used to authenticate	Unauthorized administrative access	MFA required, short-lived tokens, break-glass audit
T4: Cross-tenant spoofing	Tenant A attempts to access Tenant B resources	Data breach across tenants	Separate SPIFFE trust domains per tenant, namespace isolation

T - Tampering

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T5: Container image tampering	Malicious image pushed to registry	Compromised workloads	Cosign signature verification, immutable tags, Gatekeeper admission
T6: IaC tampering	Unauthorized Terraform changes	Infrastructure misconfiguration	GitOps with PR review, branch protection, signed commits
T7: Config drift	Direct console changes bypassing IaC	Compliance violations	AWS Config monitoring, drift auto-remediation, alert on manual changes
T8: Audit log tampering	Attacker modifies logs to hide activity	Loss of forensic evidence	CloudTrail log integrity validation, S3 object lock, separate log account
T9: Air-gap package tampering	Malicious modification during transfer	Compromised air-gapped environment	OPS signatures, manifest checksums, chain of custody

R - Repudiation

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T10: Denied administrative actions	Admin claims they didn't perform action	Accountability gaps	CloudTrail with KMS signature, session recording for sensitive operations
T11: Denied data access	User claims they didn't access data	Compliance violations	Comprehensive audit logging, correlation IDs, immutable audit trail
T12: Denied policy changes	Changes made without attribution	Security control bypass	Git history for policy as code, OPA decision logs, approval workflows

I - Information Disclosure

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T13: Secrets exposure	Secrets baked in logs, env vars, or git	Credential compromise	External Secrets Operator, no secrets in IaC, log scrubbing
T14: Cross-tenant data leak	Tenant A accesses Tenant B data	Data breach	Network policies, separate KMS keys, RBAC enforcement
T15: Metadata exposure	Cloud metadata service (IMDS) exploitation	Credential theft	IMDSv2 required, pod identity limits, no host network
T16: Transit data exposure	Network sniffing	Data interception	mTLS everywhere, no plaintext protocols, VPC endpoints
T17: Storage data exposure	Unencrypted data at rest	Data breach	KMS encryption mandatory, CloudHSM for 8/9, policy enforcement

D - Denial of Service

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T18: Control plane DoS	Overwhelming EKS API server	Cluster unavailable	API rate limiting, regional HA, autoscaling
T19: Resource exhaustion	Pod consumes excessive CPU/memory	Node instability	Resource quotas, limit ranges, priority classes
T20: Network DoS	Traffic flood to services	Service unavailable	WAF, Shield Advanced, NACLs, rate limiting
T21: Edge disconnection	Network cut to air-gapped site	Operations halted	30-day offline capability, local state persistence, store-and-forward

E - Elevation of Privilege

THREAT	ATTACK VECTOR	IMPACT	MITIGATION
T22: Container escape	Kernel exploit from container	Host compromise	Seccomp profiles, AppArmor, no privileged containers, Falco monitoring
T23: IRSA escalation	Overly permissive role trust policy	Unauthorized cluster access	Least privilege IRSAC, no cluster-admin for workloads, audit regular
T24: IAM role escalation	AssumeRole to higher privilege	Cloud account compromise	Permission boundaries, SCPs, IAM Access Analyzer
T25: Break-glass abuse	Emergency access misused	Security control bypass	Dual-approval, time-limited, comprehensive audit, immediate review

Attack Tree: Supply Chain Compromise

```
Goal: Execute malicious code in production
    OR
    ├── 1. Compromise source code
    │   ├── 1.1 Compromise developer workstation
    │   ├── 1.2 Compromise CI/CD credentials
    │   └── 1.3 Malicious PR merged (social engineering)
    OR
    ├── 2. Compromise build infrastructure
```

```
+-- 2. Compromise build process
|   +-- 2.1 Inject malicious dependencies (SCA bypass)
|   +-- 2.2 Compromise build agent
|   +-- 2.3 Modify build scripts
|
+-- 3. Compromise image registry
|   +-- 3.1 Push malicious image (stolen creds)
|   +-- 3.2 Overwrite existing tag
|   +-- 3.3 Compromise registry infrastructure
|
+-- 4. Bypass admission control
|   +-- 4.1 Exploit Gatekeeper vulnerability
|   +-- 4.2 Deploy to namespace without policies
|   +-- 4.3 Use allowed but vulnerable base image
|
+-- 5. Air-gap transfer compromise
    +-- 5.1 Modify package during transfer
    +-- 5.2 Substitute malicious media
    +-- 5.3 Compromise signing key
```

Mitigations Summary

LAYER	PRIMARY MITIGATIONS
Identity	SPIFFE/SPIRE, mTLS, MFA, short-lived credentials
Supply Chain	Signed images, SBOM, vulnerability scanning, admission control
Network	Microsegmentation, private endpoints, egress filtering
Data	Encryption at rest (KMS/HSM), encryption in transit (TLS 1.3)
Audit	Immutable logs, correlation IDs, tamper detection
Edge	Offline capability, signed packages, chain of custody