# Security Requirements & Threat Model

## Purpose and Scope

This document defines the security requirements and threat model for PineCone Pro Supplies’ Enterprise Resource Planning (ERP) / Inventory Management System (IMS). It covers the end‑to‑end solution from the Angular‑based user interface through the ASP.NET microservices, containerised workloads running on Azure Kubernetes Service (AKS), and the data stores that support procurement, inventory management, order fulfilment, returns handling and financial reporting.

The goals are to:

* Protect sensitive data (e.g., customer information, financial records, hazardous‑material classifications) throughout its lifecycle.
* Ensure only authorised users and services can access system resources.
* Maintain the integrity and availability of the ERP/IMS under normal and adverse conditions.
* Comply with relevant regulations (e.g., tax laws, data privacy, hazardous‑materials handling, PCI DSS) and support auditing and forensic investigations.

## Security Objectives and Principles

The system’s security posture is built on the following principles:

1. **Confidentiality and least privilege** – access to data and functionality must be restricted to those who need it to perform their job. The ERP security best practices article stresses that robust access management and the principle of least privilege mitigate insider threats【502724759256560†L118-L124】.
2. **Integrity** – prevent unauthorised modification of data and code. Tampering risks undermine data accuracy and can lead to incorrect orders, payments or tax filings.
3. **Availability and resilience** – the system must remain available to authorised users while withstanding denial‑of‑service (DoS) attacks and hardware failures. A resilient network architecture and disaster‑recovery plan provide barriers against cyber threats【502724759256560†L165-L177】【502724759256560†L241-L252】.
4. **Auditability and accountability** – record all relevant events to allow reconstruction of actions. Comprehensive logging and monitoring enable prompt detection of anomalies and support compliance【502724759256560†L214-L225】.
5. **Defense in depth** – security controls must exist at multiple layers (application, network, platform, human) to prevent single points of failure.
6. **Secure by design and continuous improvement** – embed security considerations early through threat modelling and update the model and controls as the system evolves.

## System Overview and Trust Boundaries

The ERP/IMS uses a microservices architecture. Users interact with an Angular single‑page application served via an API gateway (ingress controller). Each backend service runs as a containerised ASP.NET Core process within an AKS cluster, exposing REST/GraphQL APIs and subscribing to asynchronous events. Core data is stored in Azure SQL Databases, Cosmos DB, and Redis caches. External integrations include payment gateways, shipping carriers, vendor EDI and a tax calculation service.

Major trust boundaries include:

* **User devices** → **Frontend (Angular)** – authentication, session management and cross‑site scripting (XSS) prevention.
* **Frontend → API Gateway** – TLS encryption, token propagation and rate limiting.
* **API Gateway → Microservices** – service‑to‑service authentication using managed identities and internal certificate‑based communication.
* **Microservices → Data stores** – network restrictions and encryption at rest for databases and caches【365799377372705†L401-L404】.
* **Microservices → Third‑party services** – secure integration over HTTPS using API keys or OAuth tokens.

The AKS nodes reside in a private virtual network with no public IPs; network policies limit pod‑to‑pod traffic【365799377372705†L439-L468】. Secrets are injected via Kubernetes Secrets and stored in memory, not on disk【365799377372705†L487-L505】.

## Security Requirements

### 1 Authentication and Access Control

1. **Role‑based access control (RBAC)** – assign users to roles with the minimum privileges required. Regularly review and adjust permissions to reflect job changes【502724759256560†L118-L124】. Leverage the existing RBAC matrix for functional roles (e.g., purchasing lead, warehouse associate) and enforce separation of duties (SoD) for sensitive actions (e.g., entering and approving purchase orders).
2. **Multi‑factor authentication (MFA)** – require MFA for all user accounts. The ERP security guide notes that MFA adds an additional layer of protection against credential theft and phishing【502724759256560†L181-L193】.
3. **Single sign‑on and identity federation** – integrate with Microsoft Entra ID to centralise identity management. Use managed identities for pods and services to access Azure resources.
4. **Session management** – enforce secure session cookies, idle timeouts and re‑authentication for sensitive operations. Prevent session fixation and cross‑site request forgery (CSRF) attacks with appropriate tokens.
5. **Service‑to‑service authorization** – use service principals and Azure Managed Identities to authenticate microservices. Limit each service’s permissions using least privilege and network policies.

### 2 Data Confidentiality and Encryption

1. **Encryption at rest and in transit** – data stored on Azure Managed Disks is automatically encrypted at rest【365799377372705†L401-L404】. Sensitive data (customer details, payment tokens, hazardous‑material classifications) must be encrypted in databases and caches. Use TLS 1.2+ for all communications, including service‑to‑service and external integrations.
2. **Secret management** – store credentials, keys and connection strings in Kubernetes Secrets and Azure Key Vault. Secrets are only delivered to pods that require them and reside in memory (tmpfs)【365799377372705†L487-L505】. Enable encryption at rest for secrets stored in etcd using customer‑managed keys【365799377372705†L513-L515】.
3. **Data classification and masking** – classify data according to sensitivity (public, internal, confidential, restricted). Apply masking or tokenisation for restricted data in logs or lower environments. Limit query access to personally identifiable information (PII).

### 3 Network Security and Segmentation

1. **Private cluster and virtual network** – deploy AKS nodes in a private subnet without public IP addresses; restrict SSH access【365799377372705†L390-L394】. Use Azure VPN or ExpressRoute for on‑premises connectivity.
2. **Network security groups (NSGs) and firewalls** – define inbound and outbound rules at subnet level to allow only required ports. Avoid modifying NIC‑level NSGs managed by AKS【365799377372705†L449-L459】.
3. **Ingress controller and API gateway** – terminate TLS and enforce rate limiting, IP whitelisting, and WAF (Web Application Firewall) rules at the edge. For internet‑facing services, use a WAF to protect against OWASP Top 10 attacks.
4. **Pod network policies** – restrict pod‑to‑pod communication to only those services that need to talk to each other【365799377372705†L463-L468】. Deny traffic to critical system components (e.g., admin databases) from non‑trusted namespaces.
5. **Isolated compute options** – for workloads requiring higher isolation (e.g., handling hazardous‑material data), consider using kernel‑isolated or confidential containers to ensure memory encryption and isolation【365799377372705†L417-L433】.

### 4 Application and Code Security

1. **Secure software development lifecycle (SSDLC)** – follow secure coding practices, including input validation, output encoding, and parameterised queries to prevent SQL injection and XSS. Use static and dynamic application security testing (SAST/DAST) and automated dependency scanning.
2. **API security** – implement strong authentication for API calls (OAuth2, JWT). Use schema validation, rate limiting, and object‑level access checks. Document APIs and ensure they do not expose sensitive fields by default.
3. **Patch management** – apply security patches promptly. ERP security guidance warns that delaying patches increases the risk of exploitation【502724759256560†L195-L207】. Automate image rebuilds and rolling updates in AKS.
4. **Configuration management** – follow vendor recommendations for secure configuration【502724759256560†L126-L137】. Disable unused services and enforce strong password policies.

### 5 Container and Kubernetes Security

1. **Least privilege for containers** – avoid running containers as root; use Pod Security Policies or built‑in admission controllers to enforce this. The AKS documentation advises limiting containers to necessary actions and processes【365799377372705†L480-L485】.
2. **Image scanning and provenance** – scan container images for vulnerabilities and sign images to verify provenance before deployment. Use Azure Container Registry Content Trust or Notary.
3. **Runtime protection** – enable Microsoft Defender for Containers or equivalent to detect and restrict attacks on pods【365799377372705†L470-L477】.
4. **Host security** – harden the host OS with updated kernels, enable seccomp and AppArmor profiles to restrict system calls【365799377372705†L480-L485】.

### 6 Logging, Monitoring and Auditing

1. **Comprehensive logging** – capture authentication attempts, privilege escalations, configuration changes, data access events and administrative actions. ERP best practices highlight the value of detailed logs for detecting anomalies【502724759256560†L214-L225】.
2. **Centralised log management** – forward logs to Azure Monitor and Application Insights. Integrate with a Security Information and Event Management (SIEM) system to correlate events and generate alerts.
3. **Audit trails and change logs** – maintain immutable audit records for compliance. The regulatory compliance guide notes that ERP systems should provide audit trails and change logs as part of their compliance features【802701672353026†L131-L139】.
4. **Real‑time monitoring and alerting** – implement dashboards and alerts for KPIs and security events. Monitor resource utilisation to detect DoS attacks or misbehaving processes.

### 7 Compliance and Regulatory Controls

1. **Data protection regulations** – adhere to GDPR, CCPA and other privacy laws by implementing data subject rights (e.g., right to be forgotten) and cross‑border transfer safeguards. The compliance article stresses the importance of data privacy and security obligations【802701672353026†L103-L115】.
2. **Financial and tax compliance** – ensure accurate financial reporting (SOX, GAAP) by enforcing segregation of duties, logging financial transactions and maintaining immutable ledgers. Automate tax calculations and record county‑level tax data as required by the primary feature list.
3. **Industry regulations** – support hazardous‑material reporting, chemical lot tracking, and vendor compliance as needed for regulated products. Provide robust audit trails for shipments and returns.
4. **PCI DSS and payment security** – store payment tokens securely, avoid storing full card numbers and use tokenisation with a PCI‑compliant payment gateway. Implement quarterly vulnerability scans and annual penetration tests.

### 8 Backup, Disaster Recovery and Business Continuity

1. **Regular backups and off‑site storage** – schedule automated backups of databases and application state. ERP best practices recommend storing backups in secure offsite locations or cloud services with encryption【502724759256560†L228-L237】.
2. **Disaster recovery plan** – maintain a disaster‑recovery runbook with recovery time objectives (RTO) and recovery point objectives (RPO). Regularly test failover to secondary regions【502724759256560†L241-L252】.
3. **Resilient deployment** – deploy services in multiple availability zones; use horizontal pod autoscaling and cluster autoscaler to handle load spikes.

### 9 People and Process Controls

1. **Security awareness training** – educate employees on phishing, social engineering, password hygiene and incident reporting. Ongoing training fosters a security culture and reduces human errors【502724759256560†L254-L266】.
2. **Incident response** – establish a response team, define procedures for detection, containment, eradication and recovery, and conduct regular drills.
3. **Offboarding protocols** – promptly revoke departing employees’ access to the ERP system to mitigate insider threats【502724759256560†L270-L279】.
4. **Third‑party risk management** – assess the security posture of vendors (e.g., payment processors, shipping providers). Use contracts and security questionnaires to ensure they meet equivalent security standards.

## Threat Model

### Methodology

We adopt the **STRIDE** threat modeling framework. STRIDE was created by Microsoft to classify threats into six categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege【555388495253724†L250-L277】. It uses data flow diagrams to systematically identify threats and encourages proactive mitigation【555388495253724†L288-L293】. Each category maps to a violation of a core security property (e.g., spoofing violates authentication)【555388495253724†L269-L277】.

### Assets

* **User identities and credentials** – employee accounts, customer login details, API keys.
* **Financial data** – invoices, payments, purchase orders, tax records.
* **Inventory data** – product SKUs, stock levels, lot/serial numbers, hazardous‑material classifications.
* **Business process data** – order statuses, vendor performance metrics, RMA dispositions.
* **System infrastructure** – container images, secrets, configuration files, underlying OS and network.

### Entry Points and Attack Surfaces

* Public web interface (Angular app) and REST APIs.
* API gateway/ingress controller and load balancers.
* Inter‑service messaging (Event Bus, Service Bus topics/queues).
* Database connections and data stores.
* DevOps pipelines (CI/CD), container registry and version control system.
* Third‑party APIs (payment processors, shipping carriers, vendor EDI).

### Threat Analysis and Mitigations

| STRIDE Category | Potential Threats | Mitigations |
| --- | --- | --- |
| **Spoofing (Authentication)** | • Attackers use stolen credentials or bypass authentication to impersonate users or services.• Man‑in‑the‑middle (MITM) attacks on API calls. | • Enforce MFA and strong password policies【502724759256560†L181-L193】.• Use TLS for all communications and validate certificates.• Employ token‑based authentication (OAuth2/JWT) and short‑lived access tokens.• Use service principals and managed identities for service‑to‑service authentication.• Monitor failed login attempts and rate‑limit authentication endpoints. |
| **Tampering (Integrity)** | • Injection attacks (SQL/NoSQL injection) alter queries or commands.• Malicious modification of data in transit or at rest.• Unauthorized image modification in container registry. | • Use parameterised queries and input validation.• Implement integrity checks (hashes, digital signatures) on critical files and images.• Enforce RBAC on container registries; sign images and validate before deployment.• Encrypt data at rest (disks, databases)【365799377372705†L401-L404】 and in transit.• Use network policies to prevent direct access to databases【365799377372705†L463-L468】. |
| **Repudiation (Accountability)** | • Users deny performing actions, or logs are tampered with.• Inadequate logging leads to incomplete forensic evidence. | • Maintain immutable audit logs with timestamped records【802701672353026†L131-L139】.• Digitally sign critical transactions (e.g., approval of purchase orders).• Store logs in a secure, write‑once medium (e.g., append‑only storage).• Integrate logs with SIEM for correlation and alerting. |
| **Information Disclosure (Confidentiality)** | • Data breaches expose PII, hazardous‑material classifications or financial data.• Unencrypted traffic or misconfigured backups leak secrets.• Excessive permissions allow employees to view sensitive records. | • Encrypt sensitive data at rest and in transit【365799377372705†L401-L404】【502724759256560†L152-L160】.• Implement field‑level encryption or masking in the data model.• Use least‑privilege access controls and role segregation【502724759256560†L118-L124】.• Apply network segmentation and firewall rules【502724759256560†L165-L177】.• Secure backups and offsite storage【502724759256560†L228-L237】. |
| **Denial of Service (Availability)** | • Distributed denial‑of‑service (DDoS) overwhelms public endpoints.• Resource exhaustion attacks on database connections or message queues.• Application errors cause cascading failures. | • Use Azure DDoS Protection and Web Application Firewall.• Implement autoscaling and resource limits on pods.• Apply rate limiting at the ingress and API gateway.• Use circuit breakers and bulkheads in microservice interactions.• Continuously monitor health and resource utilisation; alert on anomalies. |
| **Elevation of Privilege (Authorisation)** | • Attackers exploit flaws to gain higher privileges or run arbitrary commands.• Misconfigured containers run with root privileges. | • Enforce RBAC and SoD controls; restrict high‑risk actions to specific roles【502724759256560†L118-L124】.• Use Pod Security Policies/Admission Controllers to prevent privileged containers【365799377372705†L480-L485】.• Harden host OS and disable root login inside containers.• Apply SAST/DAST scanning to detect vulnerabilities enabling privilege escalation. |

### Additional Threat Considerations

* **Third‑party integrations** – insecure APIs from vendors or payment providers may lead to data leakage or transaction tampering. Require vendors to use TLS and strong authentication; perform periodic security assessments.
* **Insider threats** – disgruntled employees could misuse their access. Mitigate via least‑privilege, real‑time monitoring, SoD and offboarding protocols【502724759256560†L270-L279】.
* **Supply chain attacks** – malicious code or dependencies inserted into containers or dependencies. Mitigate by using signed images, verifying software supply chain provenance, and scanning dependencies.
* **Physical security** – ensure physical security of Azure datacentres is handled by the provider; for on‑premises components (e.g., local cross‑dock warehouses), implement surveillance, access control and environmental controls.

### Risk Prioritisation and Continuous Improvement

Assess each threat’s likelihood and impact. Use frameworks like FAIR to assign risk scores and prioritise mitigations. The STRIDE process should be revisited whenever new features, integrations or architecture changes are introduced【555388495253724†L320-L324】. Combine design‑time threat modelling with runtime monitoring to detect zero‑day exploits【555388495253724†L397-L402】.

## Conclusion

Implementing these security requirements and maintaining a living threat model enables PineCone Pro Supplies to protect critical data, maintain compliance and deliver reliable services. By applying defense‑in‑depth controls—ranging from robust identity management and encryption through network segmentation, secure coding, container hardening, logging, and user training—the organisation can mitigate the risks associated with running a complex ERP/IMS. Continual threat modelling and security reviews ensure the architecture adapts to evolving threats and regulatory requirements【555388495253724†L288-L293】.