ATC Crisis Management System – Security Model Summary

Purpose

This document explains the **security posture and trust model** of the ATC Crisis Management System. It focuses on *why* security controls exist and *where* trust boundaries are enforced, without repeating architectural or implementation details covered elsewhere.

This document is intended for: - IT leadership - Security reviewers - Future system owners n---

1. Core Security Principles

The system is designed around the following principles:

- **Zero implicit trust**: No request is trusted solely because it reached a flow endpoint
- **Defense in depth**: Multiple, layered checks protect every sensitive action
- **Fail closed**: If validation cannot be completed, processing stops
- **Least privilege**: Access is limited to the minimum required scope

---

2. Primary Trust Boundary: Microsoft Entra ID

Microsoft Entra ID is the authoritative identity provider for the system.

What Entra ID guarantees: - The caller successfully authenticated - The token was issued by ATC's tenant - The token has not expired

What Entra ID does **not** guarantee: - That the account is still active - That the account is authorized for crisis submission - That the token has not been replayed or forwarded

Because of these limitations, **Entra authentication alone is insufficient**.

---

3. Token Validation Strategy

Why Tokens Are Parsed Inside Power Automate

The system does not blindly trust the presence of a bearer token.

Instead: - The JWT payload is decoded - The caller's Object ID (OID) is extracted - The OID is used for downstream validation

This ensures: - The identity used for authorization is explicit - Authorization decisions are not based on client-supplied metadata

---

4. Active Account Verification (Critical Control)

Why This Exists

Azure AD tokens can remain valid even if: - A user is disabled - A user leaves the company - A device is compromised but token remains cached

Enforcement

After token decoding: - Microsoft Graph is queried using the caller's OID - The user's **accountEnabled** state is checked

If the account is not active: - Processing stops immediately - A **403 Forbidden** response is returned

This prevents: - Former employees submitting incidents - Stolen tokens being reused - Disabled accounts interacting with the system

---

5. Authorization Beyond Identity

Authentication answers *who the user is*. Authorization answers *what they are allowed to do*.

Additional controls include: - Explicit flow-level authorization conditions - Header validation for internal HTTP calls - SharePoint permissions scoped to service accounts

This ensures: - Internal flows cannot be triggered externally - Cross-flow calls cannot be spoofed

---

6. Graph API Usage and Risk Containment

Microsoft Graph is used for: - User validation - Teams messaging - Leadership notifications

Risk is mitigated by: - Narrowly scoped app permissions - Secret-based authentication - Short-lived access tokens

If Graph access fails: - Processing halts or safely degrades - No partial or unauthenticated actions are taken

---

7. Why Custom Headers Still Exist

Even with Entra ID authentication: - Internal HTTP-triggered flows include shared secret headers

This provides: - Protection against accidental exposure of endpoints - A second validation factor between internal components

These headers are **not** user-facing and are rotated as part of operational maintenance.

---

8. Data Protection Considerations

- No credentials are stored in the client
- Photos are compressed before transmission
- Sensitive data resides only in:
- SharePoint (controlled access)
- Microsoft Teams (internal visibility)

Local browser storage is used only for: - Temporary offline persistence - Non-sensitive workflow state

---

9. Security Philosophy

This system assumes: - Tokens can be stolen - Accounts can be disabled mid-session - Network calls can fail

Security controls are intentionally redundant.

If a security check cannot be completed, the system **refuses to proceed**.