Fleet Inventory Specification

Status: 🚧 Draft - Pending validation
Approach: Tag-based grouping and relationships

Defines the **SIEM fleet** of a client (where to deploy configuration).

Core Concept: Tags

Everything is tagged. Tags define:

- **Cluster membership**: `cluster: production`
- **Node role**: `role: storage`, `role: search`, `role: aio`
- **Relationships**: `sh-for: production` (SH sees specific DNs)
- **Environment**: `env: prod`, `env: staging`
- **Custom grouping**: `site: berlin`, `tier: gold`

Global Structure

```
# fleet.yaml
apiVersion: cac-configmgr.io/v1
kind: Fleet

metadata:
  name: client-alpha

spec:
  managementMode: director

  director:
    poolUuid: "aaa-1111-bbb-2222"
    apiHost: "https://director.logpoint.com"
    credentialsRef: "env://DIRECTOR_TOKEN_ALPHA"

  # All nodes are flat - tags define relationships
  nodes:
    # AIOs
    aios:
      - name: aio-dr
        logpointId: "lp-aio-dr-01"
        tags:
          - env: dr
          - role: aio
          - site: disaster-recovery

    # Data Nodes
    dataNodes:
```

```yaml
    - name: dn-prod-01
      logpointId: "lp-dn-p1"
      tags:
        - cluster: production      # ← Cluster membership
        - env: prod
        - site: berlin

    - name: dn-prod-02
      logpointId: "lp-dn-p2"
      tags:
        - cluster: production      # ← Same cluster
        - env: prod
        - site: munich

    - name: dn-archive-01
      logpointId: "lp-dn-arc"
      tags:
        - cluster: archive         # ← Different cluster
        - env: prod
        - tier: cold-storage

    - name: dn-legacy-site
      logpointId: "lp-dn-leg"
      tags:
        - env: prod
        - site: legacy             # ← No cluster tag = standalone

  # Search Heads
  searchHeads:
    - name: sh-prod-01
      logpointId: "lp-sh-p1"
      tags:
        - cluster: sh-frontend     # ← SH cluster
        - env: prod
        - sh-for: production       # ← Sees production DN cluster
        - sh-for: archive          # ← Sees archive DN cluster

    - name: sh-prod-02
      logpointId: "lp-sh-p2"
      tags:
        - cluster: sh-frontend     # ← Same SH cluster
        - env: prod
        - sh-for: production
        - sh-for: archive

    - name: sh-admin
      logpointId: "lp-sh-adm"
      tags:
        - env: prod
        - role: admin              # ← No cluster = individual
        - sh-for: production
        - sh-for: archive
        - sh-for: legacy           # ← Can see standalone DNs
```

```
      - name: sh-soc-external
        logpointId: "lp-sh-soc"
        tags:
          - env: prod
          - role: soc
          - sh-for: production       # ← Limited access
```

How Tags Work

1. Cluster Membership

Nodes with the **same** `cluster: <name>` tag form a cluster.

```
# These 3 nodes form the "production" cluster
dn-prod-01: tags: [cluster: production]
dn-prod-02: tags: [cluster: production]
dn-prod-03: tags: [cluster: production]
```

2. SH to DN Relationships

A Search Head sees Data Nodes based on tags:

Tag on SH	Meaning
`sh-for: production`	Sees all DNs with `cluster: production`
`sh-for: legacy`	Sees all DNs with `site: legacy` (or tag `legacy: true`)
No `sh-for` tag	Sees nothing (or everything if admin)

3. AIO Behavior

AIOs can participate in relationships via tags:

```
aios:
  - name: aio-backup
    logpointId: "lp-aio-bu"
    tags:
      - cluster: backup-aio        # ← Can cluster with other AIOs?
      - env: prod
      - sh-for: production         # ← As SH, sees production DNs
      - dn-for: search-cluster     # ← As DN, seen by SH cluster
```

Use Cases

Use Case 1: Simple AIO Client

```
spec:
  nodes:
    aios:
      - name: aio-main
        logpointId: "lp-aio-01"
        tags:
          - env: production
          - site: hq
```

Use Case 2: Distributed with Standalone DNs

```
spec:
  nodes:
    dataNodes:
      - name: dn-site-a
        logpointId: "lp-dn-01"
        tags:
          - site: site-a
          - env: production

      - name: dn-site-b
        logpointId: "lp-dn-02"
        tags:
          - site: site-b
          - env: production

    searchHeads:
      - name: sh-central
        logpointId: "lp-sh-01"
        tags:
          - env: production
          - sh-for: site-a          # ← Explicit reference to standalone
          - sh-for: site-b          # ← Can reference multiple
```

Use Case 3: Full Cluster (Bank)

```
spec:
  nodes:
    dataNodes:
      - { name: dn-prod-01, logpointId: "lp-dn-p1", tags: [{cluster:
production}, {env: prod}] }
      - { name: dn-prod-02, logpointId: "lp-dn-p2", tags: [{cluster:
production}, {env: prod}] }
      - { name: dn-prod-03, logpointId: "lp-dn-p3", tags: [{cluster:
production}, {env: prod}] }

      - { name: dn-dr-01, logpointId: "lp-dn-d1", tags: [{cluster: dr},
{env: prod}] }
```

```
      - { name: dn-dr-02, logpointId: "lp-dn-d2", tags: [{cluster: dr},
{env: prod}] }

      - { name: dn-archive, logpointId: "lp-dn-a1", tags: [{cluster:
archive}, {env: prod}] }

    searchHeads:
      - { name: sh-01, logpointId: "lp-sh-01", tags: [{cluster: frontend},
{sh-for: production}, {sh-for: dr}] }
      - { name: sh-02, logpointId: "lp-sh-02", tags: [{cluster: frontend},
{sh-for: production}, {sh-for: dr}] }

      - { name: sh-admin, logpointId: "lp-sh-adm", tags: [{role: admin},
{sh-for: production}, {sh-for: dr}, {sh-for: archive}] }

      - { name: sh-soc, logpointId: "lp-sh-soc", tags: [{role: soc}, {sh-
for: production}] }
```

Use Case 4: Prod + Staging (Same Fleet)

```
spec:
  nodes:
    dataNodes:
      - name: dn-prod-01
        logpointId: "lp-dn-p1"
        tags:
          - cluster: production
          - env: prod

      - name: dn-staging-01
        logpointId: "lp-dn-s1"
        tags:
          - cluster: staging
          - env: staging

    searchHeads:
      - name: sh-prod
        logpointId: "lp-sh-p1"
        tags:
          - env: prod
          - sh-for: production

      - name: sh-staging
        logpointId: "lp-sh-s1"
        tags:
          - env: staging
          - sh-for: staging
```

Tag Conventions

Reserved Tags (CaC-ConfigMgr interprets these)

Tag	Usage	Example Values
`cluster`	Group nodes into clusters	`production`, `archive`
`env`	Environment separation	`prod`, `staging`, `dev`
`sh-for`	SH visibility scope	Cluster names or other tags
`role`	Special roles	`admin`, `soc`, `dr`

User-Defined Tags (Free form)

```
tags:
  - site: berlin
  - tier: gold
  - cost-center: it-security
  - compliance: pci-dss
```

Benefits of Tag Approach

1. **Flexibility**: Add any metadata without schema changes
2. **Multi-dimensional**: A node can be in multiple logical groups
3. **Selector-friendly**: `cac-configmgr apply --select env=prod,tier=gold`
4. **Future-proof**: New relationship types via new tag conventions
5. **Familiar**: Same pattern as Kubernetes labels, AWS tags, etc.

Design Decisions

Q3 - AIO Clustering: ✅ YES

AIOs CAN form clusters via tags. Use case: DRP (Disaster Recovery Plan) with resilient AIO clusters.

```
aios:
  - name: aio-primary
    logpointId: "lp-aio-p1"
    tags:
      - cluster: drp-ha          # ← Same cluster = same config
      - site: primary

  - name: aio-secondary
    logpointId: "lp-aio-p2"
    tags:
      - cluster: drp-ha          # ← Same cluster
      - site: dr

  - name: aio-standalone
```

```
    logpointId: "lp-aio-s1"
    tags:
      - site: branch            # ← No cluster tag = unique
```

Q5 - Tag Validation: PERMISSIVE (recommended)

Any tags allowed. Reserved tags have special meaning.

Reserved tags (interpreted by CaC-ConfigMgr):

- `cluster`: Groups nodes into clusters
- `env`: Environment separation (prod, staging, dev)
- `sh-for`: SH visibility scope
- `role`: Special roles (admin, soc, dr)

User tags (free form, for filtering/organization):

- `site`, `tier`, `cost-center`, `compliance`, `owner`, etc.

Tag Conflicts Resolution

If a node has conflicting tags (e.g., two `cluster` values), LAST tag wins or ERROR (configurable).

```
# Valid - multiple user tags
tags:
  - cluster: production         # ← Reserved: cluster membership
  - site: berlin                # ← User tag
  - compliance: pci-dss         # ← User tag
  - owner: security-team        # ← User tag
```