AGENT — Log Engineering & Parser Auto-Generator Agent

Full Architecture Blueprint (MSSP-Ready, Multi-Tenant, Enterprise Grade)

---

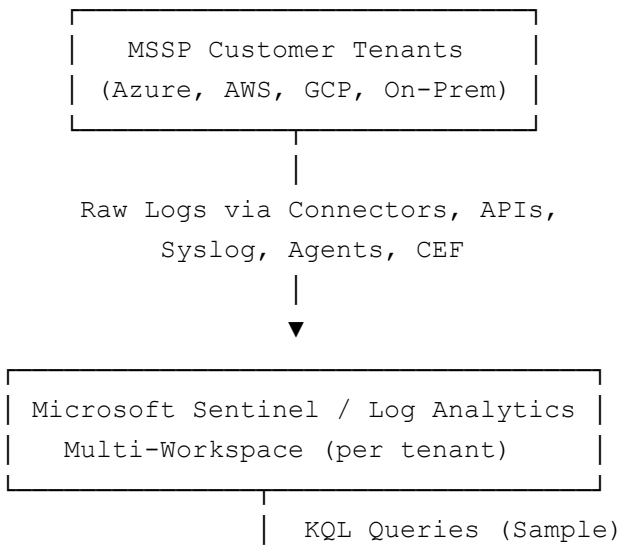
1. Purpose & Mission

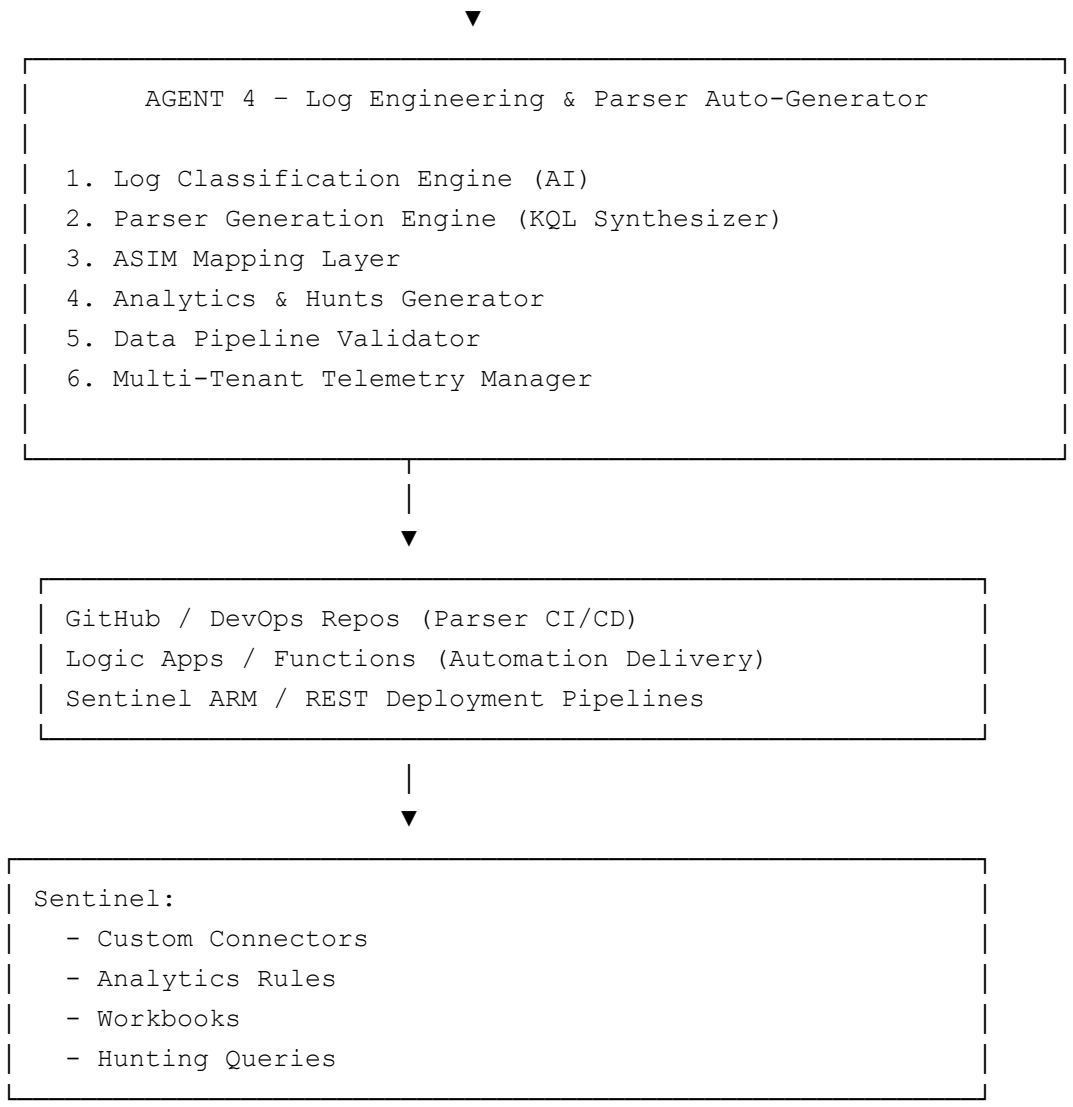
This agent automates everything related to log onboarding and normalization across an MSSP environment:

- Parse raw logs → generate optimized KQL parsers
- Identify missing fields → generate ASIM-compliant schemas
- Auto-create Sentinel analytic rules, hunting queries, and workbooks
- Detect ingestion failures and auto-heal connectors
- Provide multi-tenant engineering insights for CPX CRS operations

---

2. High-Level Architecture Diagram (Text Representation)





---

3. Detailed Component Architecture

3.1 Log Classification Engine

The AI uses structured + unstructured models to identify:

- Log source type (firewall, EDR, IAM, cloud activity, proxy, OT, syslog, custom)
- Vendor & product (Fortinet, CheckPoint, Palo Alto, CrowdStrike, Okta, SAP, etc.)
- Event category & schema
- Required transformation for Sentinel ingestion

Tech Stack

- Azure OpenAI model (GPT-4/5)
- Python-based classifier hosted in Azure Functions
- Optional: Defender for Cloud Apps for SaaS footprint enrichment

---

3.2 Parser Generation Engine (KQL Synthesizer)

This module takes raw log samples and auto-generates:

Outputs

- KQL parser function
- Extraction logic using regex, parse_json, split
- DataType mapping field-by-field
- Time normalization logic

Example Output

```
let Cisco_ASA_Parser = (raw: string) {
  parse raw with TimeGenerated:datetime " " Device string ":" Action string ...
  | project TimeGenerated, Device, Action, SourceIP, DestinationIP, RuleName
};
```

Automation Flow

1. Agent uploads raw log sample to the engine
2. Engine generates draft parser
3. Linter validates:
    - Correct KQL syntax
    - ASIM naming standards
    - Required columns exist
4. CI/CD pipeline pushes to GitHub

3.3 ASIM Mapping Layer

The agent aligns every parser with the **Advanced Security Information Model (ASIM)**:

ASIM Outputs Generated

- Normalized schema
- Data type mapping
- Event type classification
- MITRE mapping
- Normalization table function

ASIM Types Supported

- NetworkSession
- DnsActivity
- Authentication
- ProcessEvents
- HttpActivity
- CloudSecurityPosture

3.4 Analytics & Hunts Generator

Based on the parser + ASIM output:

Generated Content

- NRT & Scheduled analytics rules
- MITRE mapped detections
- Hunting queries
- Rule tuning recommendations (thresholds, suppression)

Example Analytics Rule

```
{
  "displayName": "Suspicious Admin Login from New Country",
  "severity": "High",
  "query": "... KQL generated by agent ...",
  "tactics": [ "CredentialAccess", "InitialAccess" ]
}
```

3.5 Data Pipeline Validator

Continuously monitors telemetry across tenants:

Checks

- Log connector ingestion failures
- Sudden drop/spike in event volume
- Missing mandatory ASIM fields
- Latency between source → LA workspace

If Failure Detected

- Auto-ticket in ServiceNow
- Optional: self-heal
    - restart connector
    - re-push connector's configuration
    - replay missing logs

3.6 Multi-Tenant Telemetry Manager

Enables MSSP-wide monitoring:

- Watches all Sentinel workspaces through Lighthouse
- Aggregates ingestion by:
    - Customer
    - Region
    - Log type
    - ASIM family
- Generates MSSP dashboards:
    - "Top Failing Parsers"
    - "Ingestion Gaps"
    - "Missing Data by Tenant"
    - "Parser Coverage Heatmap"

---

4. Automation, APIs, and Integrations

Microsoft Stack Integrations

- **Microsoft Sentinel REST API**
  - /dataConnectors
  - /alertRules
  - /automationRules
- **Log Analytics API**
  - Query ingestion behavior
  - Validate sample log structure
- **Azure Resource Manager**
  - Deploy saved ARM templates for much faster onboarding
- **Logic Apps**
  - Automate parser deployments
  - Ship hunting queries into each workspace
- **GitHub/ADO CI/CD**
  - code linting
  - parser versioning
  - automated publishing

---

5. Data Flow (Step-by-Step)

Step 1 — Customer raw logs arrive

Logs enter through:

- Syslog/CEF
- AMA extension
- Custom connector
- API ingestion

Step 2 — Agent samples logs

The agent fetches a 1–5 min sample via Log Analytics API.

Step 3 — Engine generates parser

AI synthesizes structured KQL parser.

Step 4 — Validation

CI/CD pipeline runs:

- Syntax validation
- ASIM compliance checks

Step 5 — Deployment

Logic App pushes parser → Sentinel workspace(s)

Step 6 — Analytics & Hunts auto-created

The agent deploys:

- analytics
- hunts
- watchlists
- lookups

Step 7 — Continuous monitoring

Multi-tenant telemetry service monitors performance.

---

6. Security Architecture

Identity & Access

- Agent uses Managed Identity
- RBAC:
    - Reader on customer tenant workspaces
    - Contributor on internal MSSP engineering workspace
    - GitHub OIDC for CI/CD

Data Handling

- No raw logs stored locally
- Data encrypted at rest (Azure default)
- CLS: Customer data stays in tenant boundary

Governance

- Parser changes → pull request → mandatory review
- Change history stored in code repo
- Monthly compliance report on log coverage

---

7. Output Artifacts per Customer

The agent produces:

Artifact	Description
KQL Parser	ASIM-compliant parser function
Normalization Table	Mapping of all key fields
Analytics Rules	MITRE-mapped detections
Hunting Queries	Investigation-ready hunts
Workbooks	Parser and ingestion health dashboards
CI/CD Pipelines	Automated deployment templates
Documentation	Markdown tech spec

---

8. Benefits for MSSP

- 80–90% faster onboarding of new log sources
- Consistent parser quality across analysts
- Automated ASIM adoption
- Auto-healing ingestion improves SLA
- Engineers spend less time writing KQL manually
- Multi-tenant telemetry reduces blindspots