



Monitoring the great value in your Big Data

PRODUCT INTRODUCTION

TrackMe for Splunk

The key concepts behind the product — what TrackMe is, what it tracks, how it decides, and how it tells you.



In a large Splunk environment, data silently breaks.



A forwarder stops

The index keeps ingesting from other hosts. The dead machine is invisible.



A sourcetype arrives late

Detections run on a window that no longer contains the events they need.



A field stops parsing

Correlation searches quietly return less — or nothing. Nobody notices.



A scheduled search skips

Summary indexes and reports drift. Scheduler debt accumulates over years.

Splunk itself will not tell you.

The searches simply return less, or nothing, and nobody notices until a detection misses or a dashboard goes blank. Modern Splunk estates are not tidy lists of feeds: thousands of indexes, tens of thousands of sourcetypes, hundreds of thousands of hosts. You cannot watch — or even classify — all of that by hand.

What is TrackMe?

TrackMe is a Splunk app that discovers, maintains, and monitors the availability and quality of your data — across one Splunk deployment or many, at any scale.

Think of it as a control plane on top of Splunk: it continuously watches raw index activity and turns it into something you can operate on — per-entity health states, SLOs, stateful alerts and, optionally, LLM-generated investigations.

The questions classic Splunk monitoring leaves open



Is my data still flowing?

Every data source, host and metric feed is tracked for delay, latency and volume — and licensed volume per index.



Is my data still good?

Field parsing quality and CIM compliance are monitored, not assumed.



Are my searches still healthy?

Scheduled-search execution, skipping and errors are first-class entities.



Who do I tell, and how urgently?

Priority, SLA and impact scoring decide what is worth an alert and what is noise.

The mental model in one sentence

TrackMe tracks entities (grouped into components, isolated inside virtual tenants); on every cycle a decision maker gives each entity a state and a score, policies classify it, and alerting turns critical states into notifications.

1 Entity

The unit TrackMe monitors: one data source, host, metric feed, scheduled search or custom object. Has an identity, a state, a score and metadata.

2 Component

The kind of entity — DSM, DHM, MHM, VOL, FLX, FQM or WLK. Decides what is measured and what “healthy” means.

3 Virtual Tenant

An isolated monitoring domain: one customer, environment or team. Own entities, config, roles — the unit of isolation and scale.

4 Decision maker

The shared engine that, each cycle, reads an entity’s metrics and computes its state and score — one consistent answer everywhere.

5 Policy

Assigns priority, tags and SLA automatically by rule (regex, lookup or SPL). Policies suggest; a manual override always wins.

6 Alerting

Stateful: one incident record per problem — opened, updated, closed — delivered by email, notable event, ingest or command.

Seven components — grouped by the question they answer

Is the data arriving?

DSM

Data Source Monitoring
index × sourcetype feeds

DHM

Data Host Monitoring
host × sourcetype activity

MHM

Metric Host Monitoring
hosts sending metric data

delay · latency · volume

Is the volume normal?

VOL

Volume Outliers
licensed volume per index

Is the licensed volume of each index normal? ML outliers on drops and spikes.

Is it well-formed?

FQM

Field Quality Monitoring
parsing quality & CIM

Is every field parsed and CIM-compliant? Coverage and success measured per field.

Are searches healthy?

WLK

Workload Knowledge
scheduled-search health

Are searches skipping, erroring or running late? Execution health as first-class entities.

Anything else?

FLX

Flex Objects
any SPL → tracked entity

Everything else — you write the SPL; it becomes an entity with custom KPIs and dynamic thresholds.

Plus UAM — User Activity Monitoring (Beta): a dedicated tenant type that watches Splunk accounts and their activity; findings from a policy catalogue, no entities.

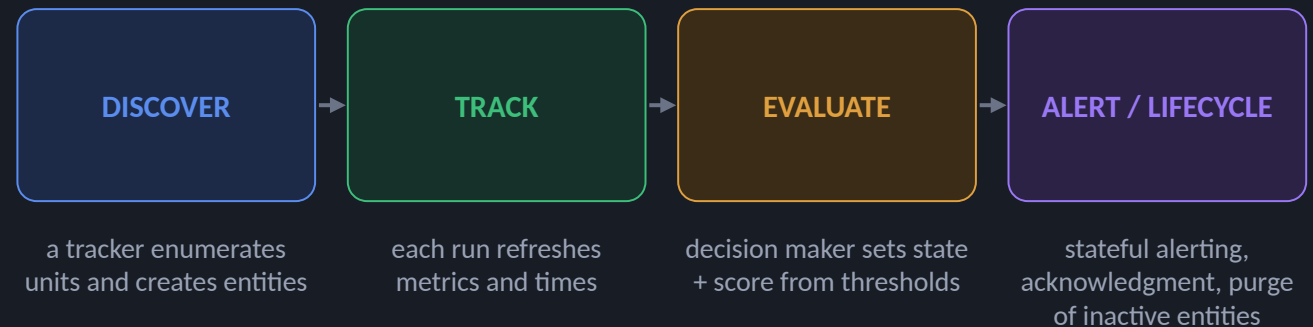
Seven components, one entity model

However different the seven components look, every entity has the same shape and the same lifecycle. Learn one component — you have half-learned them all.

Every entity always has

object	its canonical, unique key — what it identifies differs per component
alias	an editable display name
state	green · orange · red · blue
score	the impact score the state derives from
thresholds & metrics	anomaly fields and a cache of the latest measurements
metadata	priority, tags, labels, SLA class

...and goes through the same lifecycle each cycle



Trackers are scheduled searches, usually running every five minutes — the freshness of a delay measurement is only as precise as the tracker cadence. Entities can also be declared before they exist (expected sources / hosts): a declared entity starts red until real data arrives.

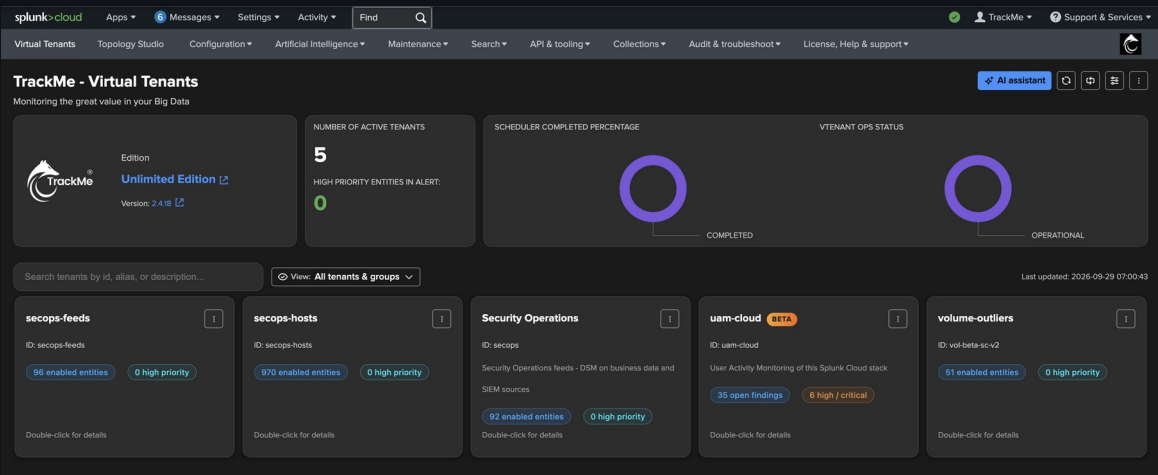
Virtual Tenants — the unit of isolation and the unit of scale

A virtual tenant is a self-contained, isolated instance of TrackMe inside your Splunk app. It owns its data, configuration, trackers, access control and — optionally — its own remote Splunk deployment to monitor.

Why tenants exist

- **Segmentation** by company, country, business unit or team
- **By technology** firewall, performance and application logs with their own thresholds
- **By scope** one tenant per Splunk cluster, index family or search-head tier
- **Access control** teams own their data spaces without seeing each other's entities

Virtual Groups aggregate entities from several tenants into one read-only cross-tenant card (NOC view). Remote deployments: a tenant can monitor the local search head or a remote Splunk Enterprise / Cloud over a token-rotated connection with failover — the experience is identical.



The Virtual Tenants page — the entry point of the product

How TrackMe works — the monitoring cycle

TrackMe is not a live dashboard that queries Splunk when you open it. Trackers run on a cycle and persist their results; the UI reads the last computed state. This is what lets it scale to thousands of entities and alert without anyone watching a screen.



Steps 3–6 are the heart of the product — the rest of this session is a tour of each step.

Entity state and impact score



Healthy

green

No active anomaly. The vast majority of entities should be green most of the time.



Warning

orange

A non-critical anomaly is active — score above zero, below critical. An early signal, not yet paging.



Critical

red

The entity has crossed the critical threshold. The alert fires — by design.



Suppressed

blue

Protected from alerting — group protection, disruption grace or maintenance — even though an anomaly may be present.

Hybrid impact scoring — additive, predictable, explainable

total_score	=
base_score	from the metrics index
+ impact_score_delay	if the delay threshold is breached
+ impact_score_latency	if the latency threshold is breached
+ impact_score_outliers	from ML outlier detection
+ threshold_scores	sum of breached dynamic thresholds
+ ...	every contribution is a configurable weight

The score maps to a colour



= 0



< 100



≥ 100

Weights are configurable per tenant and can be overridden per entity — you decide whether a latency breach matters more than a volume drop. Nothing happens without a traceable reason.

Classify what matters. Protect against noise.

Classification metadata

Priority

critical / high / medium / low / pending — how much an entity matters

Tags

free-form string metadata for filtering and grouping

Labels

a curated, colour-coded catalogue of chips, assignable by rules

SLA class

platinum / gold / silver with associated thresholds

Policies do it automatically

Setting priority or SLA on thousands of entities by hand does not scale. A policy assigns metadata by rule — a regex, a Splunk lookup or an SPL search. Policies suggest; a manual override always wins; removing a policy auto-cleans what it assigned.

Why a red score might not page you



Logical group protection

N entities judged collectively. If enough members are green, struggling members are demoted to blue — one flaky member of a healthy cluster does not page you.



Disruption grace

A new anomaly is held in blue until it has persisted for a minimum time — transient blips are absorbed, only sustained problems escalate.



Maintenance mode

The entity, tenant or whole instance is inside a maintenance window.

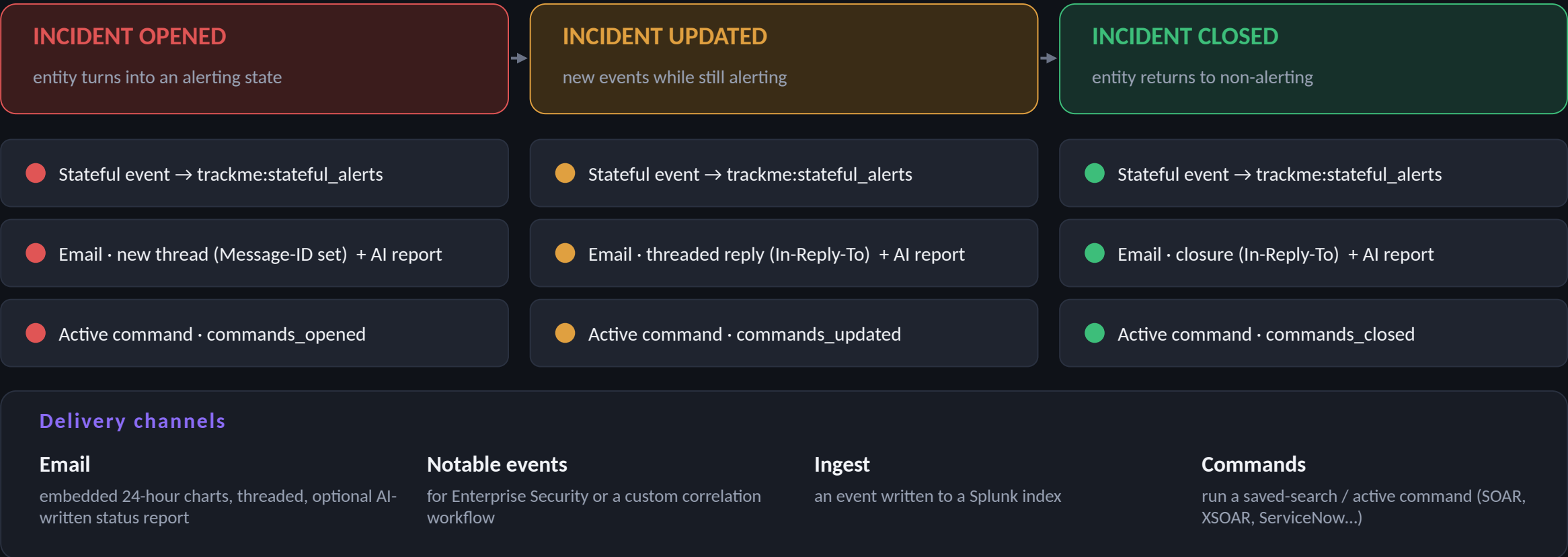


Acknowledgment

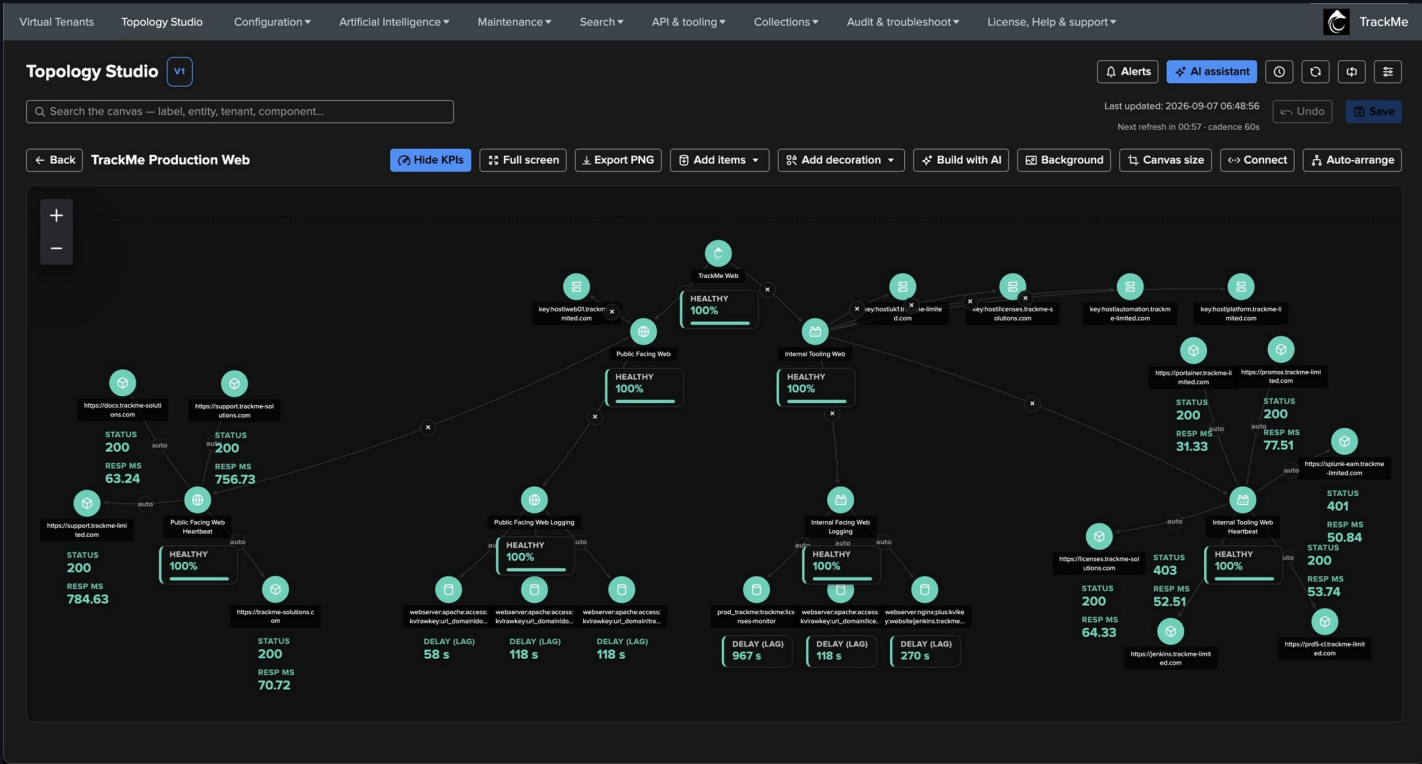
Someone already accepted this problem. Sticky until cleared, or unsticky with automatic expiry; revoked if the anomaly reason changes.

Stateful alerting — one incident per problem

Instead of re-firing every cycle an entity is in trouble, TrackMe keeps one incident record per problem: it opens when the entity first goes into alert, updates while the problem persists, and closes when the entity recovers. Rich, context-aware notifications without the spam.



Topology Studio — your view of the environment, kept alive



Build live dependency maps from real TrackMe entities — place them, connect them, share them, watch their status update in real time. And because a map you cannot act on is just a picture, any topology can also alert.

Four node types

- Entity** one monitored entity, live state colour, optional on-canvas KPI
- Aggregate** live roll-up of many entities (scope + filter) — worst state, per-state counts
- View** another saved view folded into one health node — a map of maps
- Label** free-text annotation for zones, tiers or flows

Built from real entities

Authored by you

Shared under RBAC

Alerting-capable

AI-integrated

Available in every edition — Foundation, Enterprise, Unlimited and trials — including full authoring and Topology Alerts. Complementary to the built-in per-component topology graphs (auto-derived, read-only) and Virtual Groups (cross-tenant counters, no canvas).

TrackMe Agentic — AI alongside the monitoring cycle

The AI layer helps operators understand and resolve what TrackMe detects, grounding every answer in the live state of the environment and keeping the operator in the loop. Built in four levels — they read best in that order.

Level 1

AI Assistant

The conversational front door — a context-aware chat drawer on every entity, page and feature. Understands your environment; can write AI status reports into alert emails.

Level 2

AI Advisors & Concierge

Six specialists that inspect and, with consent, act: ML models, feed lifecycle, FLX thresholds, field quality, component health, UAM findings — plus the Concierge generalist.

Level 3

AI Agents automation

The same advisors running unattended on a nightly schedule, scoped by per-tenant policy, within guardrails you define.

Level 4

AI Routines

Your own standing instructions in natural language: describe what to watch for, TrackMe evaluates it on your schedule, then notifies or acts under the consent you granted.

Opt-in only

Completely inert until an administrator configures a provider. No data leaves, no LLM call is made, until you opt in — from air-gapped Ollama to Splunk-hosted LLMs.

Trust

Agents never make up entity state: every claim traces to a tool call; outputs are typed; read-only inspect mode is the default.

Auditability & traceability

Runtime guardrails and hard caps enforce the blast radius; every tool call and every write is indexed and prefixed [AI Agent] in the audit trail.

How TrackMe is built — three layers, all inside Splunk

Presentation

A modern React interface — Virtual Tenants, Tenant Home, Topology Studio, dashboards — and the optional AI Assistant.

Control plane

A REST API and a per-tenant KV Store hold all configuration and current state — the single source of truth for the UI, the alerts and the API.

Data plane

Scheduled searches (trackers) run on a cycle and a shared decision maker computes each entity's state and score. This is where "is it healthy?" is decided.

You do not need to know the internals to use TrackMe — but the shape explains why it behaves the way it does: the UI reads state, it does not compute it.

Deployment facts

Runs inside Splunk

Splunk Enterprise, Splunk Cloud or hybrid. No outbound SaaS — suitable for air-gapped environments.

Local or remote

A tenant monitors the local search head or a remote deployment over an authenticated, token-rotated connection with automatic failover.

Built for scale

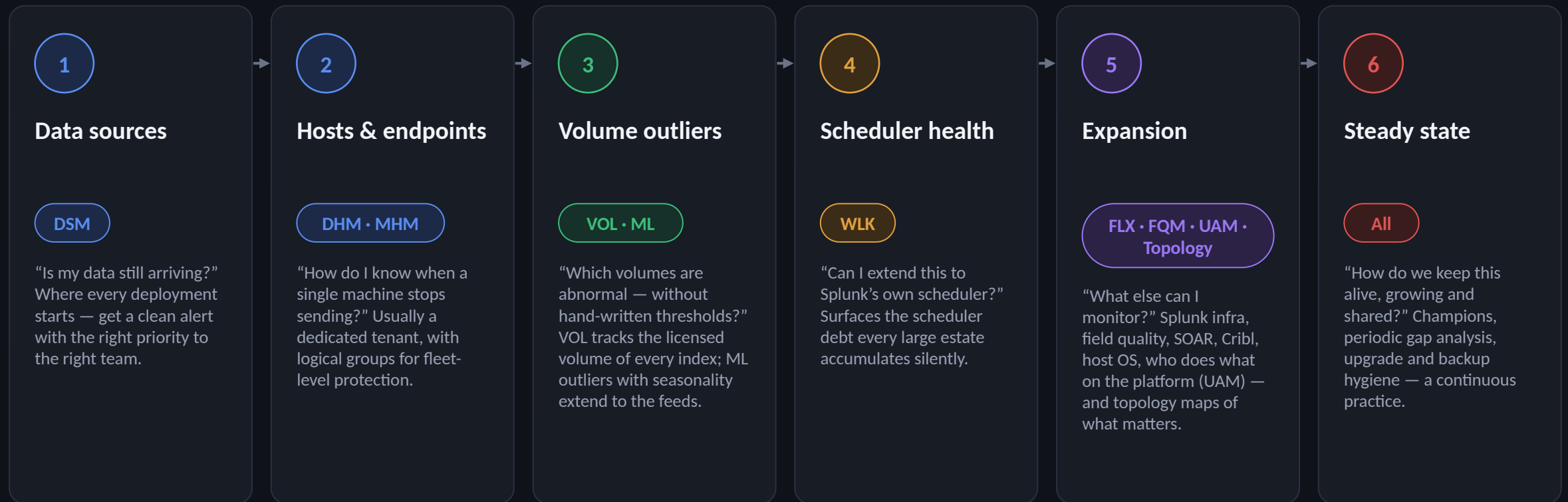
Multi-tenant, in production above 100 TB/day. Scale and cadence are tuned per tenant.

Editions

Foundation, Enterprise, Unlimited. A new install starts a 90-day trial; an expired license first switches to read-only while monitoring keeps running.

The adoption journey — a map, not a ladder

Most customers walk the same path — not because we prescribe it, but because the questions arrive in a particular order. Most are in several phases at once; the point is to know where you are and what is typically next.



Where we are

1

What TrackMe is

a Splunk app that discovers, maintains and monitors the availability and quality of your data

2

Seven components + UAM

DSM, DHM, MHM, VOL, FLX, FQM, WLK share one entity model; UAM (Beta) is its own tenant type

3

Virtual Tenants

the unit of isolation and scale: own entities, configuration and roles, local or remote Splunk

4

The cycle

discover → measure → decide → classify → protect → alert, on every tracker run

5

State and impact score

green, orange, red, blue — derived from one configurable impact score per entity

6

Classify, protect, alert

policies, protection and stateful incidents — with Topology Studio and the AI layer alongside

Remember the sentence:

TrackMe tracks entities (grouped into components, isolated inside virtual tenants); on every cycle a decision maker gives each entity a state and a score, policies classify it, and alerting turns critical states into notifications.