**Co daje Elastic Agent ponad „4×Beats”**

1. **Centralne zarządzanie (Fleet)**
   * Jedna **polityka** na wiele hostów, zdalne **enroll/unenroll**, **rollout zmian**, **upgrade** i podgląd **stanu agenta**.
   * Tajemnice w polityce (secrets), tagowanie hostów, kontrola wersji integracji.
2. **Katalog Integrations + data streams**
   * Gotowe **pakiety** z pipeline’ami, indeksami (logs-\*, metrics-\*) i dashboardami; spójne nazewnictwo (data\_stream.\*).
   * Mniej „klejenia” ręcznych modułów, szybszy onboarding źródeł danych.
3. **APM intake (serwer przyjmujący APM)**
   * Elastic Agent może hostować **APM Server** (endpoint dla agentów .NET/JS/Java itd.). Beats tego nie zapewniają.
4. **Security/EDR (Elastic Defend)**
   * Ochrona endpointów, reguły detekcji, izolacja hosta, response actions (w zależności od licencji). Beats to tylko zbieranie danych.
5. **Osquery – live i harmonogram**
   * Zdalne zapytania osquery do hostów (inwentarz, zgodność, szybkie audyty) z poziomu Kibany.
6. **Uptime/Synthetics w tym samym agencie**
   * Monitory HTTP/TCP/ICMP (i przeglądarkowe w Private Locations) bez osobnego Heartbeat na każdym hoście.
7. **Operacyjnie łatwiej**
   * **Mniej usług** do instalacji/utrzymania na hoście (jeden agent zamiast 3–4 beatów).
   * **Diagnostyka**: elastic-agent diagnostics, jednolite logi i statusy.
   * **Spójne TLS/CA** i keystore w jednym miejscu.
8. **Skalowanie i HA**
   * Możliwość postawienia wielu **Fleet Serverów** i balansowania ruchu agentów.

**Porównanie — skrót**

| **Obszar** | **4×Beats (File/Metric/Winlog/Packet)** | **Elastic Agent** |
| --- | --- | --- |
| Metryki, logi, winlogi, PCAP | ✅ | ✅ |
| **Zarządzanie centralne (Fleet)** | ❌ | ✅ |
| **APM intake** | ❌ | ✅ |
| **Security/EDR (Elastic Defend)** | ❌ | ✅ |
| **Osquery (live/scheduled)** | ❌ | ✅ |
| **Synthetics/Uptime** | 🔸*Heartbeat osobno* | ✅ *w Agencie* |
| **Integrations OOTB + data streams** | 🔸 moduły Beats | ✅ pakiety + logs-\* / metrics-\* |
| **Zdalny upgrade/unenroll** | ❌ | ✅ |
| **Diagnostyka jedną komendą** | 🔸 | ✅ |

\* 🔸 = możliwe, ale mniej wygodne lub wymagające osobnych komponentów.

**Rzeczy do świadomości**

* **Zużycie zasobów:** Agent konsoliduje role, ale pod spodem uruchamia lekkie komponenty. W praktyce bywa **podobnie** do sumy kilku Beats; zyskujesz jednak **zarządzanie i spójność**.
* **Offline/air-gapped:** Standalone działa „od ręki”. Dla Fleet-managed potrzebny jest dostęp Kibany do **pakietów integracji** (on-prem EPR lub cache).
* **Packetbeat:** na Windows dalej wymaga **Npcap**. Parasol Agenta tego nie zmienia.
* **Niszowe metryki/protokoły:** pokrycie jest bardzo szerokie, ale przy bardzo specyficznych scenariuszach warto porównać konkretny moduł/metricset/protocol w dokumentacji.

**W skrócie**

Jeśli chcesz **zbierać** dane — oba podejścia zadziałają. Jeśli chcesz **zbierać + zarządzać + rozwijać** na większą skalę (policy, upgrade, EDR, osquery, APM, synthetics), **Elastic Agent** wygrywa funkcjonalnie i operacyjnie.