ChatGPT

Ryzyka i złożoność w SaaS+AI: synteza Twoich repozytoriów i badań z oceną mediów oraz obserwatorów

Zakres materiału i metoda

Analiza łączy dwa strumienie dowodowe.

Pierwszy strumień to Twoje repozytoria GitHub, które przejrzałem w kolejności wskazanej w poleceniu (konektor GitHub): `sbom`, `writeups`, `chunk-chunk`, `swarm`, `ai_platform`, `HA2D`. W nich szukałem nie „ładnych idei", tylko miejsc, gdzie system faktycznie **zamyka pętlę sterowania** (pomiar → próg → decyzja → konsekwencja) oraz gdzie w kodzie pojawiają się praktyczne odpowiedniki problemów, o których mówi rynek: koszty zmienne, ryzyko kaskad, zawodność, bezpieczeństwo, audytowalność, integracje i lock-in.

Drugi strumień to „ocena z zewnątrz": możliwie świeże (na dziś, 10 lutego 2026) sygnały rynkowo-medialne o SaaS+AI (zwłaszcza lutowe materiały 2026 o „AI-disruption"), a także źródła normatywne i techniczne (regulacje UE, standardy bezpieczeństwa AI, SRE, FinOps/FOCUS, IEA). W tym strumieniu priorytet dałem źródłom pierwotnym i oficjalnym oraz – gdzie to możliwe – źródłom polskim.

Punkt ciężkości raportu jest analityczny: nie chodzi o streszczenie repozytoriów ani przegląd prasy, tylko o ich **konfrontację**: (a) czy Twoje artefakty rzeczywiście „widzą" to, co media i obserwatorzy uznają dziś za główne ryzyka SaaS+AI, (b) gdzie są luki i niespójności, (c) jakie działania techniczne i komunikacyjne domkną całość.

Co media i obserwatorzy uznają dziś za główne problemy SaaS+AI

Najbardziej „twardym" sygnałem z początku 2026 jest to, że dyskusja o AI w SaaS przestała dotyczyć tylko produktywności, a zaczęła dotyczyć **egzystencjalnej przebudowy modeli biznesowych** i ryzyk systemowych.

W dniach 9–10 lutego 2026 entity["organization","Reuters","news agency, global"] opisywał falę globalnej wyprzedaży akcji spółek software/services, wywołaną obawą, że szybko dojrzewające narzędzia agentowe mogą „ominąć" tradycyjne modele oprogramowania i usług (katalizatorem miał być nowy produkt prawny/„legal tool" od entity["company","Anthropic","ai company, claude"]). W relacji z 9 lutego 2026 pada teza o rynkowym „wake-up call": inwestorzy zaczęli wyceniać scenariusz, w którym część SaaS przestaje być „maszyną do powtarzalnych przychodów", bo AI skraca dystans między potrzebą a wynikiem (a więc podważa wartość warstw pośrednich). [1] Jednocześnie w dniu 10 lutego 2026 entity["company","Morgan Stanley","investment bank"] ostrzegał, że to nie jest tylko historia o akcjach: jeżeli AI realnie obniży marże i stabilność spółek software, może to przełożyć się na rynek kredytowy (m.in. z powodu koncentracji ryzyka w segmencie nisko-ratingowych pożyczek dla firm software). [2]

Medialnie to spina się z dwoma „bardziej długimi" problemami, które w 2024–2026 stale wracają:

Po pierwsze: **pricing i telemetria**. entity["organization","Bain & Company","management consulting firm"] opisuje, że AI wzmacnia rozjazd między „per-seat" (łatwe do sprzedania i prognozowania) a rzeczywistą strukturą kosztu i wartości AI, gdzie koszty inferencji/fine-tuning/R&D są zmienne i często nie mają prostego licznika „na użytkownika". Wątek kluczowy to trudność przejścia na modele hybrydowe/usage/outcome ze względu na brak telemetrii produktu i infrastruktury billingowej, a także reorganizację sprzedaży i finansów. [3] W praktyce oznacza to, że „nowy pricing" jest problemem **architektury danych i kontroli**, a nie tylko marketingu.

Po drugie: **koszt zmienny i „whales"**. entity["organization","Business Insider","digital news outlet"] opisał w 2025 mechanikę „inference whales": ciężcy użytkownicy potrafią konsumować zasoby obliczeniowe o rząd wielkości większe niż wpłacany abonament, co zmusza dostawców do limitów i przejścia na usage-based (lub przynajmniej capów). [4] To nie jest detal: to mechanizm, który uczy rynek, że agentowe workflow'y (wiele kroków, dużo tokenów, retrievery, iteracje) zachowują się jak **złośliwy test obciążeniowy** wbudowany w produkt.

Na to nakłada się warstwa infrastrukturalno-energetyczna: entity["organization","International Energy Agency","intergovernmental energy org"] prognozuje, że globalne zużycie energii przez data centers ma wzrosnąć do ok. 945 TWh do 2030 (w scenariuszu bazowym), a udział serwerów „accelerated" (napędzanych głównie przez AI) ma być dominujący w przyroście. [5] Ten sam raport był szeroko komentowany medialnie (w tym Reuters). [6] W praktyce oznacza to, że „skalowalność SaaS+AI" ma dzisiaj ograniczenia nie tylko software'owe, ale też energetyczno-sieciowe (czas budowy DC vs lead time energii). [5]

Trzeci filar obserwacji z zewnątrz dotyczy bezpieczeństwa i regulacji.

Z perspektywy security, entity["organization","OWASP Foundation","security nonprofit"] w Top 10 dla aplikacji LLM (v1.1) formalizuje ryzyka, które dobrze mapują się na SaaS: prompt injection, insecure output handling, poisoning, model DoS (czyli i awarie, i spirale kosztów), supply chain vulnerabilities, disclosure. [7] To jest ważne w Twoim kontekście, bo część Twoich repozytoriów wprost buduje „pętle sterowania" – a OWASP mówi, gdzie te pętle realnie są atakowane (wejście/wyjście modelu, koszty, łańcuch dostaw danych i komponentów).

Z perspektywy governance i etyki (w praktyce: zarządzania ryzykiem), entity["organization","NIST","us standards institute"] publikuje AI RMF 1.0 jako ramę „Govern/Map/Measure/Manage"; to jest język, którym instytucje i enterprise próbują uczynić AI audytowalnym i sterowalnym. [8] Dla generatywnej AI istnieje też profil towarzyszący (który doprecyzowuje typowe ryzyka i praktyki). [9]

Wreszcie: Unia Europejska. W AI Act mocno wybrzmiewa wątek śledzenia zachowania systemu w czasie (logi, dokumentacja techniczna, post-market monitoring). W samym tekście aktu oraz w narzędziach Komisji widać nacisk na systemowe monitorowanie „w całym cyklu życia", z planem monitoringu jako częścią dokumentacji (dla high-risk). [10] To jest regulacyjny odpowiednik Twojej tezy „pętla sterowania musi być domknięta".

Równolegle Data Act wprowadza zmianę w tle ekonomii cloud/SaaS: ma usuwać bariery zmiany dostawcy, w tym stopniowe wycofanie opłat za switching i egress do 12 stycznia 2027. [11] To nie usuwa lock-inu w sensie architektury, ale osłabia jeden z jego najprostszych „fizycznych" mechanizmów (opłaty za transfer). To ważne, bo przesuwa nacisk z lock-inu „cenowego" na lock-in „systemowy" (integracje, formaty, procesy, model governance).

Co Twoje repozytoria robią z problemem złożoności i gdzie „dotykają” SaaS+AI

Twoje repozytoria nie wyglądają jak klasyczny „produkt SaaS”, tylko jak zestaw **artefaktów sterowania złożonością**: narzędzia bezpieczeństwa łańcucha dostaw, polityki odporności sieciowej, protokoły kontekstu, warstwa ontologiczna (9D) i meta-dokumenty o prawdzie/fikcji w LLM. To jest istotne: w świecie SaaS+AI największe awarie rzadko pochodzą z jednego błędu; częściej są skutkiem **kaskady w złożonym układzie** (sprzężenia, progi, opóźnienia, błędnie ustawione liczniki, brak bariery).

Najbardziej „inżyniersko twardy” element sterowania widzę w `sbom` – szczególnie w pipeline Jenkins (`lab/jenkins/pipeline_one.pipeline`). To jest bardzo czytelny, zamknięty mechanizm cybernetyczny:

- pipeline wymusza stabilność wykonania (np. `disableConcurrentBuilds()` i „przetrwanie” `durabilityHint('MAX_SURVIVABILITY')`), co jest sygnałem, że traktujesz to jako element infrastruktury sterowania, a nie jednorazowy skrypt;
- generujesz SBOM CycloneDX (`syft`) i skan (`grype`) wewnątrz kontenera toolbox;
- wyciągasz snapshot komponentów do porównania, pobierasz poprzedni snapshot z Elastic, liczysz deltę, a potem przechodzisz przez bramkę decyzyjną (`FAIL_ON ∈ {none, critical, high}`), która może unieruchomić wydanie (exit 10). To jest wprost „pomiar → próg → akcja”. W samym pipeline widać też ustandaryzowany identyfikator AID (env/app/version/repo/vcs_ref), co jest zalążkiem „metryki porównywalnej w czasie” – bez tego nie ma post-market monitoringu.

Ważny detal: pipeline wysyła do Elastic nie jeden event, ale *rodzinę zdarzeń* (`sbom_snapshot`, `sbom`, `scan`, `delta`, `gate`). To jest de facto prosty event-sourcing dla ryzyka supply chain – i w praktyce daje Ci funkcję „pamięci” (historia) oraz „stanu” (ostatni snapshot).

W Twojej dokumentacji `sbom` pojawia się też warstwa konceptualna: SBOM jako marker strukturalny oraz dopięcie kryptografii i dowodu pochodzenia do pętli DevSecOps; kluczowe jest, że wartość SBOM ujawnia się dopiero, gdy przestaje być „dokumentem”, a staje się elementem sterowania (progi kompozycji/podatności/zaufania). To jest spójne z obserwacją, że w AI-SaaS kończy się świat „raportów”, a zaczyna świat „gatingu”.

Drugi filar „zderzenia z realnością SaaS” to `swarm`, gdzie masz polityki odporności w warstwie mesh: `EnvoyFilter` do rate limiting i `DestinationRule` do circuit breaking/outlier detection. W `rate-limit.yaml` ustawiasz `failure_mode_deny: true` i timeout na 0.25s dla zewnętrznej usługi limitującej. Semantyka `failure_mode_deny` w Envoy jest fundamentalna: to wybór „fail closed”, czyli w razie problemów z usługą rate-limit proxy ma nie przepuszczać ruchu. [12] To jest bardzo mocna decyzja architektoniczna: w świecie AI (koszt na request, ryzyko przeciążenia, „whales”) fail-open często oznacza spiralę kosztów i awarię.

W `circuit-breaker.yaml` stosujesz outlier detection (5xx errors, interval 1s, ejection 30s, ejection up to 100%). Mechanika outlier detection (kiedy host wypada z puli, jak długo, jaki procent) jest standardowo opisana w dokumentacji service mesh. [13] W ujęciu SRE to jest element „odcięcia kaskady”: izolujesz źle działający upstream, żeby nie zatruł całego systemu.

Tutaj Twoje repo dotyka tego, co Google opisuje jako redukcję kaskad: najczęstszy powód cascading failures to **overload**, a skuteczną bronią jest load shedding i kontrola kolejek/limitów. [14] Zwracam uwagę: Twoje mesh-policies są wprost implementacją tej logiki, tylko na poziomie gateway/upstream.

Jednocześnie `swarm` pokazuje też „ciemną stronę" złożoności, która jest szczególnie ważna w SaaS+AI: łatwo zbudować kontrolę na brzegu (mesh), ale mieć podatne elementy w środku. Przykład: `aggregator/aggregator.py` tworzy nowy wątek dla *każdego* pakietu UDP i wysyła request HTTP bez jawnego timeoutu; w warunkach bursta to może być wektor przeciążenia i degradacji (czyli dokładnie to, co SRE identyfikuje jako początek kaskady). [14] Jeśli do tego dołożysz AI-komponenty, analogiczny pattern („jeden request → wiele kroków agentowych") jest definicją cost explosion.

Trzeci filar to `HA2D`, czyli protokół kontekstu i kontrola integralności pamięci (UUID + SHA256, timestamp, operacje store/retrieve/latest). To jest „mini-system audytu" dla stanu kontekstowego: umożliwia wykrywanie naruszeń integralności i daje ślad w czasie. To ma bezpośrednie przełożenie na wymogi audytowalności (AI Act naciska na logi i monitoring cyklu życia) [15] oraz na praktykę bezpieczeństwa (WORM-like myślenie o dowodzie).

Czwarty filar to `chunk-chunk`, gdzie w protokole HMK9D formalizujesz kontrakt `H(s)=g(F(s))`, ryzyko `R(F,g)` i „energię" jako koszt lokalny oraz globalny, a także bramki (Próg–Przejście) i „energy_guard" w procesach. To jest w istocie język kontroli złożoności: zamiast mówić „system jest skomplikowany", opisujesz go jako serię kroków Δ, gdzie każdy ma koszt i wektor w przestrzeni osi (9D). W praktyce to jest Twoja własna wersja tego, co teoria systemów nazywa pracą na sprzężeniach i progach.

Piąty filar to `ai_platform`, w którym próbujesz zmapować heterogeniczny ekosystem repozytoriów i artefaktów na spójny rejestr współrzędnych (`/QV9D`). To jest podejście do złożoności organizacyjnej: zamiast „kolejnych folderów i wiki", masz ambicję uzyskać **odwracalne mapowanie** między warstwą semantyczną a kodem. Jednocześnie w samym specyfikowaniu algorytmu mapowania pojawia się jawnie niezidentyfikowany element („DO ZAPROJEKTOWANIA": sposób deterministycznego ID). To warto potraktować jako sygnał: rdzeń konceptu jest gotowy, ale „mechanika dowodu i spójności" nadal jest ryzykiem.

Szósty filar to `writeups`: tutaj powstaje warstwa epistemiczna (prawda vs fikcja, protokoły kontekstu, „mikrokod", P0–P3, itd.). W kontekście SaaS+AI jest to potencjalnie najbardziej niedoceniany element: jeśli system generuje język, który uruchamia działanie (narzędzia, commit, API), to różnica między „ładną narracją" a „weryfikowalnym stwierdzeniem" jest różnicą między bezpiecznym systemem a performatywną halucynacją. OWASP nazywa to m.in. insecure output handling i disclosure. [7]

Konfrontacja z krytyką i aktualną oceną: zgodności, rozbieżności, luki

Twoje repozytoria w dużym stopniu są **zgodne kierunkowo** z tym, co media/obserwatorzy uznają dziś za realne ryzyka SaaS+AI, ale zgodność jest asymetryczna: jesteś bardzo mocny w „sterowaniu strukturą i granicami" (SBOM, progi, mesh-policies, pamięć i log), a relatywnie słabszy w „sterowaniu ekonomią i semantyką wyjścia" (unit economics dla inferencji, telemetria wartości/zużycia w produkcie, polityki output-safety).

Zgodności wysokiej wagi

Najważniejsza zgodność jest strukturalna: Twoje repozytoria nie traktują AI jako „feature", tylko jako **układ sprzężeń**, który trzeba zamknąć progami.

To jest dokładnie to, co Twoje wcześniejsze badania nazywają „pętlą w pętli”: gdy rośnie koszt jednostkowy (q) i/lub wykorzystanie (u), presja rynkowa pcha cenę (P), co zmienia zachowanie klientów i może destabilizować marżę. W tej samej pracy masz formalizację progu behawioralnego klienta (`u* = P/q`) oraz marży `M = (P − q·u)·S`, co wprost opisuje, dlaczego „użytkownicy-wieloryby” są problemem: w abonamencie P jest stałe, a q·u rośnie. Ta abstrakcja pasuje do zjawiska „inference whales” opisywanego w prasie: heavy usage rozsadza model flat-fee, wymuszając capy/usage-based. [4]

Druga zgodność: Twoje narzędzia techniczne są wprost „anty-kaskadowe” (rate limit, circuit breaker, bramki). To jest spójne z klasyczną teorią progów i kaskad: Granovetter pokazał, że rozkład progów może generować drastycznie odmienne wyniki zbiorowe mimo podobnych „średnich” preferencji. [16] Watts formalizował rzadkie, ale ogromne kaskady od małych bodźców w sieciach progowych. [17] W praktyce SaaS+AI oznacza to: drobna zmiana (model, prompt, billing, limit) może uruchomić nieproporcjonalne efekty (koszt, awaria, reputacja, migracje klientów). Twoje repozytoria są zbudowane tak, jakbyś zakładał właśnie tę klasę zjawisk.

Trzecia zgodność: Twój „motyw Manhattan” (superteza) opisuje mechanikę wyścigu i kaskad w warunkach presji czasu i zasobów, ale jednocześnie wskazuje na ryzyko centralizacji decyzji, błędów i kosztów. To jest zgodne z dzisiejszą oceną rynku: Reuters opisuje „przebudzenie”, w którym AI-automatyzacja może szybko przenosić wartość z jednych warstw software do innych, generując gwałtowne repricing (luty 2026). [18]

Rozbieżności i luki, które stają się ryzykiem

Najbardziej istotne luki grupują się w czterech miejscach.

Pierwsza luka: **telemetria wartości i kosztu „na jednostkę”** w runtime AI. Bain mówi wprost, że przejście do nowych modeli cenowych wymaga telemetrii produktu i infrastruktury billingowej. [3] W Twoim stacku telemetria jest mocna po stronie bezpieczeństwa kompozycji (SBOM/scan/delta/gate), ale słabsza po stronie AI-zużycia: brakuje analogicznego „AID + event types” dla (a) tokenów/sekund inferencji, (b) kosztu per workflow, (c) budżetów per tenant, (d) efektywności (np. cost per resolved ticket / per story point / per case). Bez tego Twoja własna teza ekonomiczna („pętla w pętli”) pozostaje bardziej proroctwem niż mechanizmem sterowania.

Druga luka: **security specyficzne dla LLM jako komponentu SaaS**. SBOM-moduł dobrze adresuje supply chain w sensie bibliotek/paczek, ale OWASP Top 10 wskazuje specyficzne kategorie dla aplikacji LLM: prompt injection, insecure output handling, model DoS, training data poisoning, disclosure. [7] Mesh-policies pomagają w model DoS (na poziomie transportu), ale nie rozwiązują output-handling (np. „LLM wygenerował payload, który wykonał się w downstream”). Ta luka jest istotna, bo w Twoich writeupach masz silną intuicję „słowo → czyn”, ale w kodzie/konfiguracji brakuje równie twardych „bramek wyjścia” jak w SBOM.

Trzecia luka: **regulacyjny „post-market monitoring” jako proces, nie tylko log**. AI Act (dla high-risk) wymaga systematycznego zbierania i analizy danych o działaniu w całym cyklu życia oraz planu monitoringu jako części dokumentacji. [10] Twoje artefakty (CMM w HA2D, eventy w sbom) są bardzo dobrym budulcem, ale w repozytoriach brakuje „spięcia” w jeden formalny plan: co jest KPI ryzyka, jakie są progi, kto jest właścicielem, jak wygląda eskalacja, jak wygląda analiza interakcji z innymi systemami (wprost wspomniana w akcie). [19]

Czwarta luka: **ryzyko „mnożenia logik" i brak meta-redukcji** w implementacji. Twoja „Teza o mnożeniu logik" mówi, że do kaskad prowadzi multiplikacja warstw decyzyjnych i brak nadrzędnego mechanizmu, który redukuje i porządkuje logiki w czasie. Jednocześnie w praktycznych komponentach (np. `swarm` aggregator UDP-threads, brak timeoutów, brak autoryzacji w API) widać miejsca, gdzie logika „systemowa" nie jest jeszcze zredukowana do bezpiecznych prymitywów (kolejka, backpressure, budżet, limit). To jest ryzyko, bo AI-warstwa ma tendencję do dokładania kolejnych pętli (retry, tool-use, multi-agent), co wzmacnia dokładnie tę patologię.

Regulacje i rynek: gdzie Twoje założenia są trafne, a gdzie trzeba je zaktualizować

Twoje badania o lock-inie cenowym i mechanice egress są trafne jako opis przeszłości i bieżącej praktyki rynkowej, ale Data Act zmienia parametry gry: od 12 stycznia 2027 ma zniknąć możliwość pobierania opłat za switching i egress jako bariery migracji. [11] Twoje wcześniejsze ujęcie, że dostawcy „podnoszą opłaty lock-in" i że egress staje się bronią, pozostaje prawdziwe opisowo dla epoki przed pełnym wejściem Data Act. Natomiast strategicznie oznacza to, że lock-in przesunie się w stronę: kompatybilności, specyficznych API, formatów semantycznych, „kultury procesu", oraz (w AI) lock-inu wynikającego z danych i dopasowania modeli.

To paradoksalnie wzmacnia sens Twojego podejścia do „mapowania ontologii" (QV9D) i do protokołów kontekstu: jeżeli bariery finansowe migracji maleją, rośnie waga barier semantycznych i procesowych.

Tabela porównawcza: fragmenty kodu/dokumentacji vs. oceny mediów i obserwatorów

Fragment (repo / ścieżka)	Co faktycznie robi (skrót techniczny)	Jaki problem SaaS+AI adresuje	Co mówią media/ obserwatorzy (najnowsze)	Ocena zgodności i ryzyko
`sbom/lab/jenkins/pipeline_one.pipeline`	SBOM+scan, delta względem poprzedniego snapshotu, gate (`FAIL_ON`) i blokada wydania	Sterowalność ryzyka supply chain; twarde progi („fail pipeline")	Rynek mówi: potrzebne nowe progi i telemetria, bo AI burzy stabilność modeli (luty 2026, Reuters) [1]	**Wysoka zgodność**. Ryzyko: obejmuje software dependencies, nie model/dane.
`sbom/.../pipeline_one.pipeline` (event types)	`sbom_snapshot`, `sbom`, `scan`, `delta`, `gate` w Elastic	Historia, audyt, porównywalność w czasie	AI Act akcentuje logi i monitoring cyklu życia (post-market) [20]	**Kierunkowo zgodne**, ale brakuje formalnego planu monitoringu.

Fragment (repo / ścieżka)	Co faktycznie robi (skrót techniczny)	Jaki problem SaaS+AI adresuje	Co mówią media/ obserwatorzy (najnowsze)	Ocena zgodności i ryzyko
`sbom/kryptologia-informacyjna-sbom.md`	Ujęcie: SBOM jako marker + cybernetyka „pomiar→próg→akcja” + dowód pochodzenia	Bezpieczeństwo jako sterowanie, nie raport	NIST AI RMF: Map/ Measure/ Manage jako sterowanie ryzykiem [8]	**Silna zgodność** (filozofia + praktyka).
`swarm/.../rate-limit.yaml`	Envoy rate limit, `failure_mode_deny: true`, timeout 0.25s	Ochrona przed przeciążeniem i spiralką kosztów; fail-closed	OWASP: Model DoS i kosztowe DoS jako typowy wektor [7]; SRE: overload jako źródło kaskad [14]	**Wysoka zgodność**. Ryzyko: rate limit service staje się elementem krytycznym (jego awaria = 500). [21]
`swarm/.../circuit-breaker.yaml`	Outlier detection (ejection) po 5xx	Izolacja awarii upstream; redukcja kaskad	SRE: kaskady i load shedding; outlier jako praktyka „odcinania” awarii [22]	**Wysoka zgodność**. Ryzyko: agresywność (interval 1s, 100% ejection) może generować false positives przy fluktuacjach.
`swarm/aggregator/aggregator.py`	Wątek per UDP packet; POST do API bez jawnego timeout; brak backpressure	Ryzyko „wewnętrznego DoS” i kaskady w środku systemu	SRE: overload zwykle zaczyna się od przeciążenia i kolejek; brak limitów pogarsza sytuację [14]	**Niezgodność/ ryzyko**: komponent podważa intencję mesh-policies.
`swarm/aggregator-api/aggregator_api.py`	Prosty endpoint do DB bez auth	Ryzyko nadużyć i eksfiltracji	OWASP: disclosure + supply chain + output handling jako klasy ryzyk [7]	**Luka bezpieczeństwa** (jeśli to ma być SaaS-like).

Fragment (repo / ścieżka)	Co faktycznie robi (skrót techniczny)	Jaki problem SaaS+AI adresuje	Co mówią media/ obserwatorzy (najnowsze)	Ocena zgodności i ryzyko
`HA2D/context_protocol.md`	Rekordy kontekstu z UUID, timestamp, SHA256; operacje store/ retrieve/latest	Integralność pamięci kontekstu; audyt	AI Act: logi i monitoring; NIST: governance i TEVV [23]	**Zgodne jako budulec**, brak integracji z całościową telemetrią produkcyjną.
`chunk-chunk/hmk9d_protocol.yaml`	Formalizm `H(s)=g(F(s))`, ryzyko `R(F,g)`, energia, bramki	Modelowanie złożoności jako pętli i progów	Teoria progów/ kaskad (Granovetter, Watts) pokazuje, że to właściwy język dla rzadkich kaskad [24]	**Kierunkowo bardzo silne**, ale wymaga „operacjonalizacji" w telemetry/ billing/runtime.
`ai_platform/platform.md`	Mapowanie repo/ artefaktów do `/QV9D`, próba odwracalnej struktury	Zarządzanie złożonością organizacyjną i semantyczną (governance)	FinOps/ rynek: rosną koszty i potrzeba spójnego języka wartości; FOCUS jako standaryzacja kosztów [25]	**Zgodne jako idea**, luka: brak deterministycznych ID i brak spięcia z realnym kosztem.
`writeups/...` (P0–P3, protokoły kontekstu)	Ramy: prawda vs fikcja, protokół kontekstu, mikrokod	Bezpieczeństwo epistemiczne (redukcja halucynacyjnego sterowania)	OWASP: insecure output handling i prompt injection to realne wektory w SaaS+AI [7]	**Zgodne konceptualnie**, brak twardych „bramek wyjścia" w kodzie produkcyjnym.

Luki, niezgodności i ryzyka: diagnoza „gdzie pętle nie są domknięte"

Twoje badania wprost przewidują, że układy SaaS+AI destabilizują się przez sprzężenia: cena–użycie–koszt, progi klienta, progi systemu, „mnożenie logik", kaskady. Z perspektywy implementacyjnej największe ryzyka to:

Brak ujednoliconej, produkcyjnej telemetrii kosztu i użycia AI. FinOps idzie dziś w stronę rozszerzania scope na AI/SaaS i nacisku na unit economics; FOCUS jest próbą normalizacji danych kosztowych między dostawcami. [26] Jednak w Twoich repozytoriach koszt jest bardziej „modelem" niż instrumentem. To grozi sytuacją, w której mechanizmy ochrony (rate limits, progi) są ustawiane „na czuja", a nie na podstawie obserwowalnego KPI.

Rozjazd między kontrolą na brzegu a kontrolą w środku. Mesh-policies (fail-closed rate limiting, outlier detection) są dojrzałe i zgodne z SRE, ale komponenty aplikacyjne w `swarm` mają wzorce, które generują przeciążenie (wątek per pakiet, brak backpressure). To jest klasyczna luka, która w systemach złożonych tworzy kaskady: kontrolujesz bramę, ale silnik się przegrzewa od środka. [14]

Brak formalnej warstwy „bramek semantycznych" dla output-driven systems. Twoje writeupy bardzo trafnie opisują, że słowo może stać się czynem, ale OWASP mówi: jeżeli output nie jest traktowany jak dane niebezpieczne, kończy się to exploitami i wyciekiem. [7] W kodzie nie widzę „policy engine", który wymusza sanity/validation wyjścia modelu (schematy, allow-listy, reguły wykonania narzędzi, sandbox).

Niedomknięty „plan monitoringu" pod AI Act. Masz elementy mechaniki (logi, identyfikatory, historia), ale brak dokumentu operacyjnego: metryki, odpowiedzialności, proces eskalacji, analiza interakcji między systemami (wprost wymagana w monitoringu, gdy relewantna). [19]

Rekomendacje techniczne i komunikacyjne: konkretne kroki domykające spójność

Poniżej rekomendacje podaję tak, by działały przy nieokreślonym budżecie: zaczynam od rzeczy o wysokiej dźwigni (duży efekt, umiarkowany koszt), potem proponuję warstwę „docelową".

Rekomendacje techniczne

Zbuduj „AI-telemetrię" analogiczną do SBOM-telemetrii: event types + historia + progi. W praktyce: stwórz odpowiednik AID dla wywołań AI (tenant, workflow_id, model_id, narzędzia, licznik tokenów, czas, koszt). Następnie: (a) łap `delta` w czasie (czy rośnie średni koszt per workflow), (b) przetwarzaj progi i budżety per tenant, (c) materializuj decyzje jako „gates" (throttle, degrade, deny, require human review). To jest bezpośrednie operacjonalizowanie Twojego wzoru `M = (P − q·u)·S` i progu `u*`.

Włącz FinOps/FOCUS jako warstwę normalizacji kosztu. Nawet jeśli nie implementujesz pełnego FOCUS od razu, warto przyjąć go jako docelowy schemat danych kosztowych, bo minimalizuje „ręczne ETL" między vendorami i scope'ami (cloud/SaaS/AI). [27] Dla Twojej architektury byłby to naturalny odpowiednik `/QV9D` tyle że dla ekonomii: jeden język danych kosztu.

Uczyń `swarm/aggregator.py` odpornym na overload: backpressure, limity, timeouts. To jest „twarde SRE": kolejka zamiast wątku per pakiet, limit równoległości, bezwzględne timeouty HTTP, retry z budżetem, a najlepiej mechanizm odrzutu. SRE wprost wskazuje: ograniczaj kolejki i odrzucaj pracę, której nie opłaca się przetwarzać, zanim system wejdzie w spiralę. [14] To jest też obrona przed „agentowym" stylem obciążenia.

Zszyj mesh-policies z metryką ryzyka i kosztu. Dziś rate limit i circuit breaker są „ślepe" na koszt ekonomiczny. Docelowo powinny działać na wielowymiarowych deskryptorach: tenant, plan, typ workflow, „koszt na krok" (Twoja „energia"), ryzyko. To jest praktyczne wdrożenie HMK9D jako runtime-policy.

Dopnij warstwę bezpieczeństwa LLM zgodną z OWASP:
- wejście: anty-prompt-injection (policy + separacja narzędzi),
- wyjście: walidacja i safe-rendering (insecure output handling),
- ochrona przed DoS (limity tokenów, limity narzędzi, limity iteracji agentów),
- supply chain: nie tylko paczki (SBOM), ale też modele, prompty, dane, retrieval sources. [7]
Twoje SBOM-podejście jest świetnym szkieletem: przenieś je na „SBOM dla modelu" (model card + data lineage + wersje promptów + narzędzia).

Zbuduj plan post-market monitoring w stylu AI Act/NIST jako artefakt repozytoryjny. Nie chodzi o zgodność legalną „na papierze", tylko o spójne sterowanie: KPI, progi, odpowiedzialności, mechanizmy reagowania i retencja danych. NIST AI RMF i jego profile są dobrym językiem, żeby to uporządkować (Govern/Map/Measure/Manage). [28] AI Act wymusza systematyczność i „monitowanie w czasie" – warto to potraktować jako architekturę, nie compliance. [19]

Rekomendacje komunikacyjne

Zamień „metafory" na publiczny język sterowania: progi, pomiary, SLA, zasady degradacji. Rynek (Bain) mówi, że pricing AI bez telemetrii jest niewykonalny, a Reuters pokazuje, że inwestorzy rozliczają firmy z tego, czy rozumieją własną ekspozycję na disruption. [29] Komunikacyjnie warto więc mówić: jakie są liczniki (u), jakie są progi (u*), co robisz przy przekroczeniu (degrade, cap, human-in-the-loop).

Przygotuj narrację „Data Act zmienia lock-in": mniej egress, więcej interoperacyjności i semantyki. W Polsce można to spinać komunikacyjnie także źródłami publicznymi (gov.pl), żeby pokazać, że „koszt migracji" przestaje być tylko kwestią cennika, a staje się kwestią architektury. [30]

Włącz jasny rejestr ryzyk AI (security/ethics/cost) i politykę wyjątków. Twoje repozytoria SBOM dobrze rozumieją wyjątki jako element sterowania (wyjątek ma właściciela, expiry, widoczność). Ten wzorzec przenieś na AI: wyjątki od limitów, od polityk danych, od zasad output-handling powinny być mierzalne (inaczej stają się „mnożeniem logik").

Priorytetowe źródła cytowań i dlaczego są kluczowe

Najwyższy priorytet (bo definiują „ramę realności" i obowiązki):

- entity["organization","European Commission","eu executive body"] / tekst AI Act i narzędzia Komisji (wymóg monitoringu/logów i cyklu życia) [10]
- entity["organization","Ministerstwo Cyfryzacji","warsaw, poland"] (polskie ujęcie Data Act, w tym zniesienie opłat za zmianę dostawcy od 12.01.2027) [31]
- entity["organization","NIST","us standards institute"] AI RMF 1.0 + profil GenAI (język zarządzania ryzykiem: govern/map/measure/manage) [28]
- entity["organization","OWASP Foundation","security nonprofit"] Top 10 LLM Apps (najbardziej praktyczna lista ryzyk aplikacyjnych LLM) [7]

Wysoki priorytet (bo tłumaczą „dlaczego system się wywraca"):

- entity["organization","Google","tech company"] SRE: cascading failures i overload/load shedding (operacyjna fizyka awarii) [14]
- entity["organization","International Energy Agency","intergovernmental energy org"]: energia jako constraint skalowania AI/data centers [32]

- Reuters-strumień lutego 2026: materializacja „AI disruption” w wycenach i ryzyku kredytowym [18]

Priorytet uzupełniający (bo przenoszą problem na „ekonomię i operacje”):

- entity["organization","FinOps Foundation","finops org"] + FOCUS spec (język kosztu i wartości w wielu scope’ach, w tym AI i SaaS) [33]
- entity["organization","Bain & Company","management consulting firm"]: telemetria jako warunek transformacji pricingu SaaS pod AI [3]
- entity["organization","Urząd Ochrony Danych Osobowych","warsaw, poland"]: sygnał polskiego kontekstu ochrony danych w zastosowaniach AI (szczególnie, gdy AI wchodzi w HR/biometrię) [34]

Diagram: gdzie Twoje „pętle” spotykają się z pętlami rynkowymi

```
flowchart TB
  A[Agentowe użycie AI rośnie] --> B[Zużycie: tokeny/czas/narzędzia rośnie]
  B --> C[Koszt zmienny q·u rośnie]
  C --> D[Presja na pricing/capy/limity]
  D --> A

  B --> E[Ryzyko overload/kaskad]
  E --> F[Rate limit + load shedding + circuit breaker]
  F --> B

  G[SBOM/scan/delta] --> H[Gate: progi]
  H --> I[Decyzja: GO/STOP/mitigacja]
  I --> G

  J[Logi + pamięć kontekstu] --> K[Post-market monitoring]
  K --> J
```

Ten diagram jest jednym zdaniem: Twoje repozytoria już mają dwa silne „hamulce” (SBOM-gate i mesh-gate), ale brakuje trzeciego hamulca, który dziś najbardziej boli SaaS+AI: **runtime-telemetrii ekonomicznej** i bramek sterowania kosztem/zużyciem per workflow/tenant. To jest dokładnie miejsce, gdzie Twoja teza o „pętli w pętli” może przestać być opisem, a stać się mechanizmem sterowania.

---

[1] [18] https://www.investing.com/news/stock-market-news/us-software-stocks-tumble-sparks-concerns-that-ai-trade-is-reshaping-markets-4493058
https://www.investing.com/news/stock-market-news/us-software-stocks-tumble-sparks-concerns-that-ai-trade-is-reshaping-markets-4493058

[2] https://www.reuters.com/business/finance/ailed-software-selloff-may-pose-risk-15-trillion-us-credit-market-says-morgan-2026-02-10/
https://www.reuters.com/business/finance/ailed-software-selloff-may-pose-risk-15-trillion-us-credit-market-says-morgan-2026-02-10/

[3] [29] https://www.bain.com/insights/per-seat-software-pricing-isnt-dead-but-new-models-are-gaining-steam/
https://www.bain.com/insights/per-seat-software-pricing-isnt-dead-but-new-models-are-gaining-steam/

4 https://www.businessinsider.com/inference-whales-threaten-ai-coding-startups-business-model-2025-8
https://www.businessinsider.com/inference-whales-threaten-ai-coding-startups-business-model-2025-8

5 32 https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai
https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai

6 https://www.reuters.com/technology/artificial-intelligence/global-trade-war-may-produce-headwinds-nascent-ai-sector-iea-says-2025-04-10/
https://www.reuters.com/technology/artificial-intelligence/global-trade-war-may-produce-headwinds-nascent-ai-sector-iea-says-2025-04-10/

7 https://owasp.org/www-project-top-10-for-large-language-model-applications/
https://owasp.org/www-project-top-10-for-large-language-model-applications/

8 28 https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10
https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10

9 https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence
https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence

10 19 https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-72
https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-72

11 https://digital-strategy.ec.europa.eu/en/policies/data-act-explained
https://digital-strategy.ec.europa.eu/en/policies/data-act-explained

12 21 https://www.envoyproxy.io/docs/envoy/latest/api-v3/extensions/filters/http/ratelimit/v3/rate_limit.proto
https://www.envoyproxy.io/docs/envoy/latest/api-v3/extensions/filters/http/ratelimit/v3/rate_limit.proto

13 https://preliminary.istio.io/latest/docs/reference/config/networking/destination-rule/
https://preliminary.istio.io/latest/docs/reference/config/networking/destination-rule/

14 22 https://sre.google/sre-book/addressing-cascading-failures/
https://sre.google/sre-book/addressing-cascading-failures/

15 20 23 https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng
https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng

16 24 https://sociology.stanford.edu/publications/threshold-models-collective-behavior
https://sociology.stanford.edu/publications/threshold-models-collective-behavior

17 https://pubmed.ncbi.nlm.nih.gov/16578874/
https://pubmed.ncbi.nlm.nih.gov/16578874/

25 27 33 https://focus.finops.org/what-is-focus/
https://focus.finops.org/what-is-focus/

26 https://www.finops.org/insights/2025-finops-framework/
https://www.finops.org/insights/2025-finops-framework/

30 31 https://www.gov.pl/web/cyfryzacja/akt-w-sprawie-danych---nowe-zasady-wymiany-danych
https://www.gov.pl/web/cyfryzacja/akt-w-sprawie-danych---nowe-zasady-wymiany-danych

34 https://uodo.gov.pl/pl/589/3207
https://uodo.gov.pl/pl/589/3207