ChatGPT

Synteza repozytoriów DonkeyJJLove z ocenami ryzyk SaaS+AI i testem hipotez asymetrii

Potrzeby informacyjne

Aby rygorystycznie odpowiedzieć na pytanie o „asymetrię" Twojej platformy względem problemów SaaS+AI (przy braku ograniczeń budżetowych/czasowych jako założeniu roboczym), muszę ustalić:
- Jakie **mechanizmy sterowania, odporności i redukcji złożoności** są rzeczywiście zaimplementowane w kodzie/specyfikacjach (a nie tylko postulowane) i gdzie dokładnie.
- Jak te mechanizmy mapują się na **aktualne, publiczne osie krytyki** SaaS+AI: koszty i zmiana modeli cenowych, awarie/skalowalność, bezpieczeństwo (LLM+łańcuch dostaw), zgodność (AI Act), etyka i odpowiedzialność.
- Czy repozytoria **jawnie adresują** te ryzyka (testowalnie, mierzalnie), czy tylko „pasują narracyjnie".
- Jak zdefiniować **metryki asymetrii** (korzyść/ryzyko/czas/koszt) i jakie dane są potrzebne do **falsyfikacji** hipotez (panelowo i Monte Carlo).
- Jakie są **luki i sprzeczności** techniczne (np. brak kontroli dostępu, brak telemetryki kosztu LLM, słabe domknięcia pętli) oraz jak je naprawić technicznie i komunikacyjnie.
- Jak ułożyć **mapę zależności ryzyk (kaskad)** tak, by dało się ją udostępnić jako jeden spójny artefakt (esej + tabela + diagramy).

Streszczenie wykonawcze

Twoje repozytoria tworzą spójną propozycję „anty-SaaS" rozumianą jako **platforma sterowania złożonością**: pomiar → próg → akcja, domykany tożsamością bytu (AID), pamięcią zdarzeń i bramkami (gate), a po stronie runtime'u – mechaniką **odporności na kaskady** (rate limiting, circuit breaking, outlier ejection) oraz – na poziomie poznawczym/organizacyjnym – protokołami kontekstu i mapą 9D (QV9D/HMK9D). W kodzie widać realne elementy sterowania (np. „koperta zdarzenia" AID i kanał zdarzeń sbom/scan/delta/gate) oraz realne elementy amortyzacji awarii (fail-closed rate limit, outlierDetection). [1] [2] [3]

Z perspektywy zewnętrznych ocen (NIST [4], AI Act, OWASP Foundation [5], FinOps Foundation [6] / FOCUS, International Energy Agency [7], Reuters [8], entity["company","Bain & Company","management consulting"]) Twoje podejście **dobrze pokrywa** trzy klasy ryzyk: (a) obserwowalność i dowodowość (logowanie, ślad decyzji, identyfikowalność bytu), (b) łańcuch dostaw i kontrola zmian (SBOM+delta+gate), (c) odporność na przeciążenia i kaskady na warstwie infrastrukturalnej (rate limit/circuit breaker). [9]

Największe zidentyfikowane luki względem współczesnej krytyki SaaS+AI są dwie:
- **Brak "twardej" ekonomiki kosztu inferencji** (telemetria kosztu per funkcja/per klient/per agent, budżety, unit cost, automatyka ograniczeń) – przy czym zewnętrzne źródła pokazują, że ogon kosztu (tzw. „inference whales") potrafi zjeść marżę i wymusić zmianę pricingu. [10]
- **Brak formalnego domknięcia wymogów zgodności i bezpieczeństwa LLM** (np. prompt injection, output handling, kontrola danych wrażliwych) na poziomie checklist/testów i polityk – co jest rdzeniem OWASP Top 10 dla LLM aplikacji. [11]

W konsekwencji: Twoje repo przybliżają „asymetrię” jako **redukcję ryzyka systemowego i kaskad** przy rosnącej złożoności, ale nie falsyfikują jeszcze w pełni krytyki ekonomicznej (pricing/cost) ani bezpieczeństwa LLM (Top10) bez wdrożenia pomiarów i eksperymentów panelowych.

Materiały dowodowe z repozytoriów

Repozytorium SBOM jako aparat sterowania zmianą i ryzykiem

W `sbom` widać architekturę, w której SBOM nie jest raportem „do przeczytania”, tylko **czujnikiem** w pętli sterowania *pomiar → próg → akcja*. Jest to nazwane wprost w kontrakcie AID: AID ma spinać „artefakty obserwacji i sterowania” (SBOM, scan, delta, gate) w jedną tożsamość bytu i umożliwić korelację w czasie. [1]

Kluczowy fragment implementacyjno-pojęciowy to „koperta zdarzenia” (`@timestamp`, `event_type`, `aid{...}`, `payload{...}`), która jest minimalnym, przenośnym formatem do indeksowania i filtracji w narzędziach obserwowalności (Elastic/Splunk). To jest ważne dla asymetrii, bo w SaaS klasycznym często brakuje *jednej* spójnej osi identyfikacji bytu i decyzji (zamiast tego są rozproszone logi, metryki i ticketing). [1][2]

W dokumentacji pipeline’u jest jawne rozdzielenie obserwacji na typy (`sbom`, `scan`, `delta`, `gate`) i wskazanie artefaktów `.lab_out/*` (SBOM CycloneDX, raport skanu, snapshot, delta, gate vars). To jest „fizyczna lista dowodów” potrzebna do sterowania zmianą i do audytu (kto, co i dlaczego zablokował). [3]

Repozytorium swarm jako amortyzator kaskad runtime

W `swarm` znajdują się twarde mechanizmy infrastrukturalne, które wprost odpowiadają na ryzyko „kaskad” przez przeciążenia:
- **Rate limiting z trybem fail-closed** (`failure_mode_deny: true`) oraz timeoutem na usługę rate-limit (0.25s). To jest szczególnie istotne w kontekście AI, bo przeciążenia często mają charakter kosztowy i obliczeniowy (LLM DoS, runaway calls). Fail-closed działa jak *bezpiecznik*: w razie utraty kontroli nad komponentem limitującym nie „otwierasz bramy” na nieograniczony ruch. [2]
- **Circuit breaking przez outlier detection** (wyrzucanie endpointów po serii 5xx, agresywne 100% ejection). To jest klasyczny mechanizm zatrzymania propagacji awarii poprzez izolację wadliwych instancji. [3]

W warstwie aplikacyjnej `aggregator.py` pokazuje wzorzec „ingest → walidacja → forward”: odbiór UDP, dekodowanie z `errors='replace'` (nie wywalaj pętli na błędnych bajtach), parser JSON, a następnie forward HTTP do API. To jest drobny, ale realny detal asymetrii: w systemach kaskadowych często padają „meta-usługi” ingestujące dane; tu kod jest napisany tak, by nie wywracać się na błędzie dekodowania i by nie blokować pętli (wątki). [4]

Jednocześnie `aggregator_api.py` ujawnia typową lukę: endpoint przyjmuje dane i zapisuje do DB, ale w widocznym fragmencie brak jest warstwy uwierzytelniania/autoryzacji (a to w SaaS+AI często jest źródłem incydentów). Jest minimalna walidacja pól i parametryzowane SQL (co akurat redukuje SQLi), ale nie rozwiązuje nadużyć/zalewu/poisoningu danych wejściowych. [5]

Repozytorium chunk-chunk jako formalizacja kompresji i „energetyki" decyzji

W `hmk9d_protocol.yaml` (repo `chunk-chunk`) wprost zapisujesz kontrakt decyzyjny jako złożenie: **H(s) = g(F(s))**, gdzie `F` jest kompresją stanu do znaku, a `g` polityką decyzji. Dodatkowo pojawia się formalizacja ryzyka `R(F,g)` i energii kroku `E(Δ)` oraz `E_total`. To jest punkt, w którym Twoja „asymetria" różni się od SaaS: SaaS zwykle modeluje system jako zestaw usług i SLA, a tu modelujesz go jako układ sterowania z kosztami lokalnymi i globalnymi oraz osiami (T,S,R,E,I,F,A,P,D). [16]

To jest istotne również dlatego, że współczesna krytyka SaaS+AI przesuwa ciężar z „feature'ów" na **koszt i ryzyko agentowości**: długie trajektorie, rekurencja, rosnące tokeny, rosnący koszt (wprost OWASP LLM04). Kontrakt HMK9D jest semantycznym szkieletem, który *da się* podpiąć pod instrumentację kosztu/ryzyka (choć w repo brak jeszcze gotowej metryki pieniężnej). [17]

Repozytoria ai_platform i HA2D jako warstwa mapowania i pamięci

`ai_platform/platform.md` definiuje mapowanie logicznego woluminu `/QV9D` na fizyczną strukturę kodu i repozytoriów: INF/SEM/MAND + mosty + typ artefaktu (SPEC/STATE/METRICS/RITUAL/CI). To jest forma „metamodelu platformy", która – jeśli jest utrzymywana operacyjnie – redukuje koszt koordynacji i drift semantyczny między modułami. To jest też sposób na to, by „złożoność" była indeksowalna (współrzędne), a nie tylko opisywana. [18]

`LAT_GLX_PROJECT_MOSAIC.MD` wiąże repozytoria rolami w obrębie całości (chunk-chunk jako rdzeń, swarm jako warstwa rojowa, HA2D jako brama Human–AI, writeups jako kronika). To jest ważne, bo asymetria często jest w praktyce „architekturą organizacyjną": kto odpowiada za co i jak to się składa w system-of-systems. [19]

W `HA2D/context_protocol.md` pojawia się **Context Memory Manager**: rekordy mają `uuid`, `timestamp`, `payload` i `sha256` (integralność). To jest wzorzec przechodzenia od „ulotnej sesji AI" do artefaktów, które można audytować i odtwarzać – czyli dokładnie tego, czego wymagają współczesne reżimy zgodności i bezpieczeństwa (traceability, post-market monitoring). [20]

W `HA2D/readme.md` podkreślona jest rola cyklu życia (pętla generacyjna), modułów analitycznych i HUD jako jedynego interfejsu. To wzmacnia tezę o asymetrii: interfejs ma być „sterownikiem", nie tylko UI. [21]

Repozytorium writeups jako empiryczna rama „protokołów kontekstu"

W `protokoly_kontekstu_chunk-chunk_facebook_case.md` opisujesz trzy poziomy interakcji: HUMAN–AI, AI–HUMAN, AI–AI oraz proponujesz formalizację „protokołu kontekstu" jako aktualizacji stanu i funkcji decyzji, z możliwością falsyfikacji poprzez porównanie do klasyfikatora bazowego i stabilność predykcji (out-of-sample). To jest ważne, bo przenosi dyskusję o ryzykach AI z poziomu „wrażenia" na poziom *mierzalnych zależności wejście→akcja*. [22]

Konfrontacja z ocenami zewnętrznymi ryzyk SaaS+AI

Ekonomia i pricing jako rdzeń ryzyka SaaS+AI

Współczesna krytyka SaaS+AI bardzo często nie zaczyna się od „AI jest niebezpieczne", tylko od „AI zmienia strukturę kosztu i wartości, więc rozsadza pricing i przewidywalność marży". Anthropic [23] i inni dostawcy agentów uruchomili falę rynkowej niepewności, widoczną w doniesieniach o globalnych

wyprzedażach spółek software/data services po demonstracjach narzędzi automatyzujących pracę (lęk o przechwycenie wartości z tradycyjnego software i usług). [24]

W warstwie mikroekonomicznej, zjawisko „inference whales" pokazuje ogon kosztu: niewielka grupa użytkowników może generować koszty inferencji wielokrotnie przekraczające abonament, co wymusza limity i zmianę modeli cenowych. [25] To bezpośrednio łączy się z OWASP LLM04 (Model Denial of Service), gdzie „atak" może być po prostu ekonomicznym przeciążeniem zasobów przez ciężkie zapytania. [17]

Bain opisuje, że odejście od per-seat w stronę usage/output/outcome jest trudne, bo wymaga telemetryki produktu, przebudowy billing/finance i „wspólnego języka" metryk między zespołami. [26] To jest dokładnie obszar, którego Twoje repozytoria jeszcze nie domykają operacyjnie: masz język mostów/energii (HMK9D), ale nie masz jeszcze warstwy metryki kosztu pieniężnego i integracji billing.

Energetyka i infrastruktura jako twarde ograniczenie

International Energy Agency [7] wskazuje, że zużycie energii elektrycznej przez data centers ma rosnąć szybko, a w scenariuszu bazowym globalnie ok. się podwoić do ~945 TWh do 2030 r., z dużą niepewnością i silną koncentracją lokalną (lokalne bottlenecks). [27] Ta perspektywa wzmacnia Twoją oś „Semantyka–Energia": energia nie jest już metaforą – staje się realnym ograniczeniem kosztowym i regulacyjnym.

To jest też miejsce na Twoje pytanie o „różne ceny prądu dla warstw klastra": dywersyfikacja kosztu energii *może* obniżyć korelację szoków kosztowych, ale nie usuwa kaskad wynikających z przeciążeń, vendor lock-in, ani ogona kosztu inferencji. W modelach fazowych jest to redukcja sprzężenia jednego parametru (koszt energii), nie likwidacja dodatnich sprzężeń w całym systemie (popyt→koszt→limity→spadek jakości, itd.). [28]

Zgodność, logowanie i post-market monitoring

AI Act wprost wymaga, by systemy wysokiego ryzyka umożliwiały **automatyczne logowanie zdarzeń przez cały cykl życia**, w zakresie adekwatnym do celu, oraz by logi wspierały identyfikację sytuacji ryzykownych i post-market monitoring. [29] Dodatkowo art. 72 nakłada obowiązek systemu post-market monitoring i planu, a także zapowiada template Komisji do 2 lutego 2026. [30] (W polskim komunikacie rządowym widać etapowe wejście w życie przepisów i nacisk na bezpieczeństwo oraz kompetencje). [31]

Twoje repozytoria są do tego „naturalnie zestrojone" na poziomie formy danych: AID + koperta zdarzenia + pamięć kontekstu z SHA256 to kompatybilny kierunek, bo tworzy weryfikowalny ślad decyzji i stanu. [12] [20] Brakuje natomiast jawnego powiązania: **jakie zdarzenia są "relevant events"**, jaka jest retencja, jak wygląda pipeline post-market monitoring i jak są zarządzane „substantial modifications" (język AI Act). To jest luka dokumentacyjno-proceduralna, nie tylko kodowa. [32]

Bezpieczeństwo LLM i łańcuch dostaw

OWASP Top 10 dla aplikacji LLM wskazuje m.in. prompt injection, insecure output handling, model DoS, supply chain vulnerabilities i leakage danych wrażliwych. [11] Twoje repo `sbom` adresuje **supply chain** w klasycznym sensie (SBOM, delta, skan, gate), a `swarm` adresuje **DoS/overload** przez rate limiting i circuit breaker. [13] [2]

Jednak OWASP Top 10 dla LLM obejmuje również klasy ryzyk typowe dla agentów (prompt injection i output handling), które nie są widoczne jako kompletne polityki/testy w analizowanych fragmentach kodu. To oznacza, że Twoje repo są obecnie asymetryczne głównie w domenie **inżynierii systemów i obserwowalności**, a słabiej w domenie **bezpieczeństwa semantycznego LLM na wejściu/wyjściu**. [11]

Ryzyko lock-in i koszty migracji

Z polskich źródeł (Ministerstwo Cyfryzacji, PARP) wynika, że Data Act ma ułatwiać zmianę dostawcy chmurowego i wygasić opłaty za switching od 12 stycznia 2027 r. Ministerstwo Cyfryzacji [33] [34] To jest „zewnętrzny wektor asymetrii": platformy, które budują koszt/ryzyko w sposób przejrzysty i przenośny (np. standardy danych, kompatybilne logi, identyfikowalność), będą miały przewagę w świecie, gdzie migracje są tańsze i częstsze.

Twoje podejście (AID jako minimalny kontrakt + event envelope) jest kompatybilne z tą presją, ale brakuje jawnego planu: jak przenosić telemetrykę, jak normalizować koszty (tu pojawia się FinOps/FOCUS) i jak uciec od vendor-specific silosów obserwowalności. [1] [35]

Falsyfikacja hipotez asymetrii i projekt testów

Hipotezy asymetrii

W „absolutnym reżimie naukowym" hipotezy muszą być falsyfikowalne. Proponuję trzy hipotezy, które są zgodne z Twoją architekturą i jednocześnie testują to, co media/eksperci krytykują:

H1: Asymetria kosztowo-awaryjna (ogon kosztu i przeciążenia).
Platforma redukuje *prawdopodobieństwo wejścia w stan kaskadowy* (outage, runaway cost, degradacja jakości) przy tym samym obciążeniu lub tej samej klasie zadań AI, w porównaniu do wdrożenia „SaaS-na-skrót".
Uzasadnienie: fail-closed rate limit + circuit breaker + bramki + separacja ingest. [2] [3] [36]

H2: Asymetria dowodowo-zgodnościowa (traceability).
Platforma zwiększa „dowodowość" działania systemu AI (kto/co/kiedy/z jaką tożsamością) i skraca czas dojścia do przyczyny (MTTR) w incydentach, w porównaniu do systemu bez AID/koperty zdarzeń/pamięci kontekstu.
Uzasadnienie: AID + event envelope + SHA256 kontekstu jako podstawy logów i rekonstrukcji. [12] [20] [32]

H3: Asymetria złożonościowa (kompresja i sterowanie).
Platforma zmniejsza liczbę „niezależnych logik" potrzebnych do sterowania systemem (redukcja entropii sterowania) poprzez formalny kontrakt H(s)=g(F(s)) i mapę QV9D, co obniża częstość błędów organizacyjnych (np. błędne ownership, błędne zależności procesów).
Uzasadnienie: HMK9D + QV9D mapowanie repo/modułów. [16] [18]

Metryki i dane wejściowe

Dla H1 (koszt/awarie):
- **Cost tail index**: udział kosztu generowany przez top 1% (lub 0.1%) klientów/zapytań/agentów (analog „inference whales").
- **Runaway rate**: odsetek runów agentowych przekraczających budżet tokenów/czasu.

- **SLO breach rate**: liczba naruszeń latency/availability na 10k requestów.
- **Economic DoS proxy**: (koszt / czas) per request; wykrywanie anomalii.

Dla H2 (dowody/MTTR):
- **Trace completeness**: % zdarzeń z kompletnym kluczem tożsamości (AID) i korelacją do decyzji gate.
- **Mean time to explain**: czas do poprawnej rekonstrukcji „dlaczego system zrobił X".
- **Auditability score**: możliwość odtworzenia sekwencji decyzji na podstawie logów i rekordów kontekstu.

Dla H3 (złożoność):
- **Coordination entropy**: liczba wyjątków ownership/nieprzypisanych alertów, liczba ręcznych eskalacji bez mapowania QV9D.
- **Change amplification factor**: ile komponentów dotyka przeciętna zmiana (delta) vs ile decyzji/akcji wywołuje.

Projekt testu panelowego

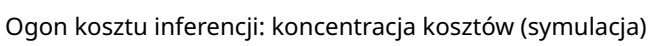
Jeżeli masz wiele projektów/usług (panel), najczystszy schemat to:
- grupa „treatment": usługi używające pełnego stosu (AID + event envelope + SBOM/delta/gate + rate limit/circuit breaker + telemetryka kosztu),
- grupa „control": porównywalne usługi bez tych mechanizmów lub z klasycznym SaaS pattern.

Model minimalny: panel z efektami stałymi (usługa i czas), outcome'y: SLO breach rate, MTTR, cost tail index. Kluczowe jest, że zewnętrzne źródła mówią o narastającej presji kosztowej i zmianie pricingu; panel ma pokazać, czy Twoja architektura zmienia nachylenie tych krzywych w czasie. [37]

Symulacja Monte Carlo jako falsyfikacja bez danych produkcyjnych

Jeśli nie masz jeszcze telemetryki kosztu, można przeprowadzić falsyfikację „wstępną" przez symulację obciążeń z ciężkim ogonem (co jest spójne z obserwacjami rynku o inference whales). [25]

Poniżej: przykładowa symulacja rozkładu kosztu (lognormalnie ciężki ogon) i koncentracji kosztów (krzywa Lorenza). To nie jest dowód empiryczny o Twoim systemie, ale jest **testem mechanizmu**: jeśli ogon jest ciężki, brak instrumentacji i limitów musi prowadzić do kryzysów pricingu lub jakości.



W tej symulacji top 1% użytkowników generuje ok. 21% kosztu (typowy wzorzec „whales"). W realnym systemie falsyfikacja H1 polega na sprawdzeniu, czy po wdrożeniu mechanizmów z `swarm` i warstwy budżetów (której jeszcze brakuje) ogon kosztu *przestaje* wymuszać spadek jakości lub zmianę pricingu.

Diagram mechaniczny: pętla sterowania i miejsca "asymetrii"

```
flowchart LR
  A[Pomiar: SBOM/SCAN/DELTA] --> B[Tożsamość bytu: AID + koperta zdarzeń]
  B --> C[Pamięć: indeks zdarzeń + kontekst]
  C --> D[Próg: gate / polityki / limity]
  D --> E[Akcja: blokada / ticket / rollout / throttle]
  E --> A
```

```
subgraph Runtime amortyzacja
  R1[Rate limiting fail-closed] --> R2[Circuit breaker / outlier ejection]
end

E --> Runtime amortyzacja
```

Ten diagram jest “mechaniczny” w sensie cybernetycznym: pokazuje, gdzie domykasz pętlę (tożsamość→pamięć→próg→akcja). AID i gate są tu Twoim rdzeniem asymetrii względem klasycznego SaaS. [1] [2] [38]

Luki, niezgodności i rekomendacje

Główne luki względem krytyki SaaS+AI

Luka kosztowa (najbardziej krytyczna): brak telemetrii i sterowania kosztem inferencji.
Źródła z rynku wskazują, że bez instrumentacji i limitów „whales” wymuszają limity i zmianę pricingu. [39]
W Twoich repo widać język „energii” (HMK9D) i mechanizmy runtime (rate limit), ale nie widać jeszcze pomostu: koszt per request/per tenant/per agent, budżety, chargeback/showback i automatyczne reguły zatrzymania w oparciu o koszt (FinOps/FOCUS). [16] [40]

Luka bezpieczeństwa LLM na wejściu/wyjściu (OWASP Top 10):
Masz mocne elementy supply-chain (SBOM) i częściowo DoS (rate limiting), ale brakuje widocznego, kompletnego „policy stack” dla prompt injection/output handling/sensitive data disclosure. [11]

Luka zgodności AI Act jako procesu (nie tylko logów):
AI Act wymaga logowania i post-market monitoring; w repo widać format danych (AID, kontekst), ale nie widać kompletnego procesu: klasy zdarzeń, retencja, review, aktualizacja planu monitoringu. [41]

Luka security-by-default w mikroserwisach:
`aggregator_api.py` w pokazanym fragmencie ma minimalną walidację, ale brak warstwy authn/authz – co w SaaS jest częstą przyczyną nadużyć i „kosztowych DoS” (zalew danych → koszty DB/IO). [15]

Rekomendacje techniczne

Najkrócej: musisz domknąć „Semantyka–Energia” w pieniądzu i w polityce.

- Zaimplementuj **telemetrykę kosztu AI** jako pierwszorzędny strumień zdarzeń (obok sbom/scan/delta/gate): `ai_usage_event` z AID + tenant + model + tokens_in/out + latency + koszt szacowany + quota/budget state. To jest warunek, by przejść na hybrydowe modele cenowe opisane przez Bain i by obronić się przed „whales”. [42] [12]
- Użyj **FinOps Scopes** (SaaS + Public Cloud + „AI”) i FOCUS 1.3 jako formatu kosztów/zużycia, aby koszty były przenośne, audytowalne i gotowe na presję Data Act/switching. [43]
- Rozszerz mechanizmy `swarm` o budżety i „policy throttling” zorientowane na koszt: rate limit nie tylko QPS, ale QPS *ważony* kosztem (np. token-em kost). OWASP LLM04 opisuje DoS także jako kosztowy/zasobowy. [17]
- Zbuduj zestaw testów/polityk dla OWASP Top10 LLM (minimum: prompt injection, output handling, data leakage) i włącz to jako osobny „gate” w Twojej pętli. [44]

- Uczyń warstwę API (np. `aggregator_api`) „secure-by-default”: authn/authz, rate limit per klient, walidacja schematu, limity rozmiaru payload, obserwowalność nadużyć. [15]
- Dla zgodności: przygotuj „szablon post-market monitoring” zgodny z AI Act (art. 72) jako artefakt platformy, a logowanie „relevant events” (art. 12) powiąż bezpośrednio z AID i z cyklem wersji. [32] [1]

Rekomendacje komunikacyjne

- Przestaw narrację z „AI platforma / anty-SaaS” na **„system sterowania ryzykiem i kosztem AI w pętli dowodowej”**. To lepiej odpowiada na język regulatora (logi, monitoring), na język FinOps (unit cost, showback), i na język rynku (przewidywalność marży przy ogonie kosztu). [45]
- W komunikacji z odbiorcami biznesowymi używaj „jednego zdania wartości”: *redukujemy prawdopodobieństwo kaskady kosztowej i awaryjnej poprzez instrumentację + progi + automatyczne akcje*. To jest dokładnie to, co media i inwestorzy adresują, obserwując paniki rynkowe wokół AI-disruption. [46]

Tabela porównawcza: fragmenty kodu vs oceny mediów/ekspertów

Repo / fragment	Mechanizm asymetrii (co robi)	Ryzyko/ocena zewnętrzna	Zgodność z krytyką	Luka / co dodać
`sbom/AID_CONTRACT.md`	Tożsamość bytu (AID) + koperta zdarzeń = korelacja decyzji i ryzyka w czasie [12]	AI Act: logowanie i traceability; NIST AI RMF: zarządzanie ryzykiem w cyklu życia [47]	Wysoka (kierunek)	Dopisać: klasy “relevant events”, retencję, plan post-market monitoring
`sbom/docs/03_JENKINS_PIPELINE.md`	Strumień sbom/scan/delta/gate + artefakty `.lab_out/*` = dowód zmian i bramkowanie [13]	NIST SSDF: praktyki bezpiecznego SDLC; OWASP supply chain [48]	Średnia-wysoka	Dodać: VEX/attestations i automatyczne policy gates
`swarm/.../rate-limit.yaml`	Fail-closed rate limit + timeout = bezpiecznik przeciw przeciążeniom [2]	OWASP LLM04 (Model DoS) + ryzyko “inference whales” [36]	Wysoka (na runtime)	Rozszerzyć na limity kosztowe per tenant/model

Repo / fragment	Mechanizm asymetrii (co robi)	Ryzyko/ocena zewnętrzna	Zgodność z krytyką	Luka / co dodać
`swarm/.../circuit-breaker.yaml`	Outlier detection i ejection = hamulec kaskad 5xx [3]	Media: rosnąca obawa o awarie/ skalowalność i utratę przewagi software [49]	Średnia	Uzupełnić o SLO-driven autoscaling i testy chaosowe
`swarm/aggregator/aggregator.py`	Odporność ingest (decode replace, nieblokowanie pętli) [14]	OWASP LLM04: rekurencja/ rozrost pracy jako forma DoS [17]	Średnia	Dodać: backpressure, kolejki, limity payload
`swarm/aggregator-api/aggregator_api.py`	Minimalna walidacja pól + zapis do DB [15]	OWASP: nadużycia, data poisoning, brak kontroli dostępu [44]	Niska	Authn/authz, schema validation, rate limit + abuse detection
`chunk-chunk/hmk9d_protocol.yaml`	Formalizacja H(s)=g(F(s)), osie 9D, energia i ryzyko [16]	NIST AI RMF (GAI profile): potrzeba profili ryzyka i mierników w cyklu [38]	Średnia (model)	Zmapować "energia" na realny koszt (tokens/PLN/ kWh) i dołączyć testy
`ai_platform/platform.md`	QV9D jako mapa systemu (INF/ SEM/MAND + mosty + artefakty) [18]	Bain: potrzeba wspólnego języka metryk (telemetry, pricing) [26]	Średnia	Zmaterializować: narzędzia do automatycznego tagowania danych i kosztów w QV9D
`HA2D/context_protocol.md`	Pamięć kontekstu (uuid/timestamp/ sha256) = integralność i rekonstrukcja [20]	AI Act: logi i monitoring; NIST: traceability i governance [50]	Wysoka (kierunek)	Polityki prywatności, retencja, minimalizacja danych, audyt dostępu

Repo / fragment	Mechanizm asymetrii (co robi)	Ryzyko/ocena zewnętrzna	Zgodność z krytyką	Luka / co dodać
`writeups/...facebook_case.md`	Empiryczna falsyfikowalność "protokołu kontekstu" (model czarnej skrzynki) [22]	Debata publiczna: systemy AI jako nieprzejrzyste, wymagające audytu [51]	Średnia	Ustandaryzować jako procedurę testów i raportowania (nie tylko narracja)

Źródła priorytetowe

Najważniejsze źródła (w kolejności "nośności" dla tez o ryzykach i asymetrii):
- NIST [4] : AI RMF 1.0 i profil dla GAI – podstawowy język ryzyk i cyklu życia. [52]
- AI Act (art. 12 i 72) + polski komunikat o wejściu w życie: logowanie, post-market monitoring, wymagania procesu. [41]
- OWASP Foundation [5] : Top 10 dla LLM aplikacji i LLM04 Model DoS – mapa praktycznych ryzyk. [11]
- FinOps Foundation [6] : Framework 2025 (Scopes) oraz FOCUS 1.3 – normalizacja kosztu i telemetryki (warunek przejścia na nowe modele cenowe). [53]
- International Energy Agency [7] : „Energy and AI" – twardy constraint energetyczny dla AI. [27]
- Reuters [8] : obserwacje rynkowe o AI-disruption jako czynniku wycen i „panik" sektorowych. [46]
- entity["company","Bain & Company","management consulting"]: zmiana pricingu i konieczność telemetryki/organizacyjnego języka metryk. [26]
- „Inference whales" jako empiryczna presja na pricing w usługach AI. [25]
- Polska perspektywa Data Act (switching fees 2027): presja na przenośność i transparentność chmury/SaaS. [54]

---

[12] AID_CONTRACT.md
https://github.com/DonkeyJJLove/sbom/blob/77fbb61e4d583ff4f91cce8c06cdc61e31201c96/AID_CONTRACT.md

[2] [6] infrastructure/istio/policies/rate-limit.yaml
https://github.com/DonkeyJJLove/swarm/blob/1fe5867fa749f376826621d6b66c0597d569bce0/infrastructure/istio/policies/rate-limit.yaml

[3] infrastructure/istio/policies/circuit-breaker.yaml
https://github.com/DonkeyJJLove/swarm/blob/1fe5867fa749f376826621d6b66c0597d569bce0/infrastructure/istio/policies/circuit-breaker.yaml

[5] [27] [28] https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai
https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai

[7] [34] [54] 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

[8] [17] [36] https://genai.owasp.org/llmrisk2023-24/llm04-model-denial-of-service/
https://genai.owasp.org/llmrisk2023-24/llm04-model-denial-of-service/

[9] [52] 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

[10] [25] [39] 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

11 44 https://owasp.org/www-project-top-10-for-large-language-model-applications/
https://owasp.org/www-project-top-10-for-large-language-model-applications/

13 docs/03_JENKINS_PIPELINE.md
https://github.com/DonkeyJJLove/sbom/blob/77fbb61e4d583ff4f91cce8c06cdc61e31201c96/docs/03_JENKINS_PIPELINE.md

14 aggregator/aggregator.py
https://github.com/DonkeyJJLove/swarm/blob/1fe5867fa749f376826621d6b66c0597d569bce0/aggregator/aggregator.py

15 aggregator-api/aggregator_api.py
https://github.com/DonkeyJJLove/swarm/blob/1fe5867fa749f376826621d6b66c0597d569bce0/aggregator-api/aggregator_api.py

16 hmk9d_protocol.yaml
https://github.com/DonkeyJJLove/chunk-chunk/blob/b71a1d69b912f3cb9d793ef433573fb7e77ae6e8/hmk9d_protocol.yaml

18 platform.md
https://github.com/DonkeyJJLove/ai_platform/blob/75dea7b28dce9fa6f9e9e37aaa514c6e04330606/platform.md

19 LAT_GLX_PROJECT_MOSAIC.MD
https://github.com/DonkeyJJLove/ai_platform/blob/75dea7b28dce9fa6f9e9e37aaa514c6e04330606/LAT_GLX_PROJECT_MOSAIC.MD

20 context_protocol.md
https://github.com/DonkeyJJLove/HA2D/blob/8b22ee8de388af3b9cf26365e1679cc6c356ef2a/context_protocol.md

21 readme.md
https://github.com/DonkeyJJLove/HA2D/blob/8b22ee8de388af3b9cf26365e1679cc6c356ef2a/readme.md

22 protokoly_kontekstu_chunk-chunk_facebook_case.md
https://github.com/DonkeyJJLove/writeups/blob/dbb82ff3e26a618fc4c86da86c3ad94c3fba94a9/protokoly_kontekstu_chunk-chunk_facebook_case.md

23 38 51 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

24 46 49 https://www.investing.com/news/stock-market-news/anthropics-new-ai-tools-deepen-selloff-in-data-analytics-and-software-stocks-investors-say-4483600
https://www.investing.com/news/stock-market-news/anthropics-new-ai-tools-deepen-selloff-in-data-analytics-and-software-stocks-investors-say-4483600

26 37 42 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/

29 32 41 45 47 50 https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-12
https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-12

30 https://artificialintelligenceact.eu/article/72/
https://artificialintelligenceact.eu/article/72/

31 https://www.gov.pl/web/cyfryzacja/pierwsze-przepisy-rozporzadzenia-o-sztucznej-inteligencji-ai-act-zaczynaja-obowiazywac
https://www.gov.pl/web/cyfryzacja/pierwsze-przepisy-rozporzadzenia-o-sztucznej-inteligencji-ai-act-zaczynaja-obowiazywac

33 35 40 https://focus.finops.org/focus-specification/
https://focus.finops.org/focus-specification/

43 53 https://www.finops.org/insights/2025-finops-framework/
https://www.finops.org/insights/2025-finops-framework/

48 https://csrc.nist.gov/Projects/ssdf

https://csrc.nist.gov/Projects/ssdf