ChatGPT

Asymetria i ryzyka SaaS+AI: konfrontacja Twoich repozytoriów z najnowszą oceną ekspertów i mediów

Potrzeby informacyjne

- Jak **zdefiniować „asymetrię”** (w SaaS+AI) w sposób mierzalny i falsyfikowalny, a nie deklaratywny.
- Które **konkretne fragmenty kodu/dokumentacji** w repo stanowią mechanizmy asymetrii (gating, telemetria, fail-closed, kontrola pamięci, „energetyka” decyzji).
- Jak te mechanizmy mapują się na **aktualne ryzyka** wskazywane przez ekspertów/standardy/rynek: kosztowe, awaryjne, bezpieczeństwa LLM, compliance (AI Act), FinOps/pricing.
- Czy Twoje repozytoria **zgadzają się z krytyką** (lub ją falsyfikują) oraz gdzie brak danych uniemożliwia rozstrzygnięcie.
- Jakie są **luki i ryzyka drugiego rzędu** (np. bramka na brzegu vs brak backpressure w środku), które mogą „zniwelować” asymetrię.
- Jak zaprojektować **testy i metryki**, które realnie rozstrzygną hipotezy asymetrii (w tym symulacje syntetyczne, jeśli brak danych z produkcji).

Artefakty asymetrii w repozytoriach

W Twoich repozytoriach widać spójny motyw: **złożoność jako układ sprzężeń**, który trzeba sterować przez „pętle” i progi (pomiar → próg → konsekwencja), a nie tylko „opisywać”. Ten motyw jest już zsyntetyzowany w Twojej wcześniejszej analizie repozytoriów.

SBOM jako asymetria „łańcucha dostaw” i twarde bramki wydaniowe

Repo `sbom` realizuje asymetrię przez **wczesne i twarde cięcie ryzyka** na etapie CI/CD. Kluczowy mechanizm to pipeline Jenkins, który materializuje cybernetykę *pomiar* → *próg* → *akcja* (SBOM+scan → próg `FAIL_ON` → `exit 10`, czyli stop wydania). Przykładowo:

```
options {
  disableConcurrentBuilds()
  durabilityHint('MAX_SURVIVABILITY')
}
parameters {
  choice(name: 'FAIL_ON', choices: ['none','critical','high'], description:
'Gate threshold')
}
...
if [ "$FAIL_ON" = "critical" ] && [ "$CRIT" -gt 0 ]; then
DECISION="STOP"; ...; fi
```

```
...
if [ "$DECISION" = "STOP" ]; then exit 10; fi
```

Asymetria polega tu na tym, że **mała reguła decyzyjna** (prosty próg) blokuje **dużą kaskadę kosztów i ryzyk** (wdrożenie podatnego artefaktu). Jest to dokładnie ten typ przewagi, który trudno skopiować organizacjom działającym w trybie „feature first": wymaga dojrzałej kultury *gatingu* i akceptacji „STOP" jako wyniku pozytywnego (ochrona).

W dokumencie `kryptologia-informacyjna-sbom.md` wzmacniasz to ujęcie epistemologicznie: SBOM jako „marker strukturalny" i **sterowanie** (nie raport). To jest ważne, bo rynek (szczególnie w AI) przesuwa się od „raportów" do „twardych bramek" i „dowodów pochodzenia".

Mesh-gating i asymetria awaryjno-kosztowa

Repo `swarm` zawiera mechanizmy, które są klasyczną odpowiedzią na awarie kaskadowe i kosztowe DoS: **fail-closed rate limiting** oraz **outlier detection/circuit breaking** na poziomie service mesh.

Rate limit (istotne fragmenty):

```
name: envoy.filters.http.ratelimit
typed_config:
  ...
  failure_mode_deny: true
  rate_limit_service:
    ...
    timeout: 0.25s
```

Wyszczególnienie `failure_mode_deny: true` jest decyzją asymetryczną: w razie awarii usługi limitującej ruch jest **odcinany**, zamiast „przepuszczany awaryjnie". To jest wprost zgodne z semantyką dokumentowaną dla Envoy (`failure_mode_deny` = nie przepuszczaj ruchu przy błędzie kontaktu z rate-limit service). [1]

Circuit breaker:

```
outlierDetection:
  consecutive5xxErrors: 5
  interval: 1s
  baseEjectionTime: 30s
  maxEjectionPercent: 100
```

To jest praktyczna implementacja outlier detection/circuit breaking w stylu Istio/Envoy. [2]
W logice SRE jest to narzędzie do hamowania kaskad, bo overload i retry-amplification potrafią zamienić małe błędy w lawinę. [3]

Jednocześnie ta sama baza kodu ujawnia ryzyko „asymetrii pozornej": bramka na brzegu i brak bramki w środku. `aggregator.py` uruchamia wątek na każdy pakiet UDP, a POST idzie bez jawnego timeoutu i bez backpressure:

```
while True:
    data, addr = sock.recvfrom(1024)
    threading.Thread(target=handle_message, args=(data, addr),
daemon=True).start()
...
response = requests.post(AGGREGATOR_API_URL, json=data_json)
```

To jest wzorzec, który w warunkach burstów może generować przeciążenie wewnętrzne (i w praktyce niwelować przewagę mesh-gatingu). Jest to spójne z obserwacją z SRE: jeśli system nie umie „degradować" i „sheddingować" w środku, kaskady wracają. [3]

Kontrakt „H(s)=g(F(s))", energia i bramki jako asymetria semantyczno-ekonomiczna

Repo `chunk-chunk` (HMK9D) formalizuje, że zachowanie systemu jest złożeniem kompresji stanu i polityki decyzyjnej:

```
behavior:
  id: "H"
  type: "S_to_A"
  definition: "H(s) = g(F(s))"
risk:
  global_risk:
    id: "R(F,g)"
energy_model:
  local_energy_symbol: "E(Δ)"
  global_energy_symbol: "E_total"
...
safety:
  energy_guard:
    metric: "E"
    max_value: 0.8
```

To jest istotne dla asymetrii, bo przenosi problem z „magii agentów" do **mierzalnych wielkości**: lokalny koszt kroku, koszt globalny, ryzyko jako spodziewana strata i bramki (progi). W świecie SaaS+AI, gdzie agentowość zwiększa liczbę kroków i tokenów, taka formalizacja jest naturalnym szkieletem do budowy polityk kosztowych i bezpieczeństwa.

QV9D jako asymetria „zarządzania mozaiką" i redukcja mnożenia logik

Repo `ai_platform` próbuje zrobić coś, co zwykle jest słabym punktem organizacji AI: utrzymać **odwracalne mapowanie** między „semantyczną architekturą" a kodem. Rdzeń brzmi: `/QV9D` jako logiczna nad-warstwa współrzędnych (warstwa/most/architektura/typ artefaktu/id). To jest potencjalnie asymetryczne, bo zamiast mnożyć wiki i foldery, mnożysz **wspólny układ współrzędnych**.

Kluczowa, krytyczna „dziura asymetrii" jest jednak nazwana wprost w dokumencie: sposób wyliczania deterministycznego `id_latarni` jest jeszcze „DO ZAPROJEKTOWANIA". Bez deterministycznych i audytowalnych identyfikatorów trudno zrobić z QV9D twardą warstwę sterowania (da się zrobić warstwę narracji).

CMM i integralność pamięci jako asymetria audytu i odpowiedzialności

Repo `HA2D` wnosi mechanizm „Context Memory Manager”: rekordy z `uuid`, `timestamp`, `payload` i `sha256` oraz operacje `STORE/RETRIEVE/GET_LATEST`. To budulec audytowalności: potrafisz wykrywać naruszenia integralności pamięci kontekstu. Jest to bezpośrednio zgodne z przesunięciem compliance w stronę logów/trace’ów oraz monitoringu cyklu życia (AI Act). [4]

„Protokoły kontekstu” jako asymetria epistemiczna i bezpieczeństwo interakcji

Repo `writeups` jest mniej „produkcyjny”, ale ma element kluczowy dla Twojego wymogu falsyfikacji: opisujesz warunek operacyjny poznania protokołu jako przewagę predykcyjną nad bazą:

> „protokół jest częściowo poznany, jeśli `acc(Ĥ) > acc_bazowa` na danych odłożonych”

To jest dokładnie most między filozofią a statystyką: wprost definiujesz, kiedy teza ma sens i kiedy ulega falsyfikacji (overfitting, drift, zła baza). To jest rzadkie w projektach AI-SaaS, gdzie tezy zwykle nie mają „kryterium śmierci”.

Porównanie z najnowszą oceną mediów i ekspertów

Obraz zewnętrzny (rynek + standardy) można streścić w jednej tezie: **SaaS+AI ma dziś trzy chroniczne ryzyka** – ekonomiczne (pricing/telemetria), awaryjno-kosztowe (agentowość i overload), oraz governance/security (LLM-specyficzne ataki + regulacje + audyt).

Ekonomia: telemetria jako warunek pricingu w AI-SaaS

entity["organization","Bain & Company","management consulting"] opisuje, że przejście od „per seat” do hybrid/usage/outcome jest trudne, bo firmy często nie mają telemetrii produktu i infrastruktury billing/finance do rozliczania AI na metrykach użycia lub wyniku. [5]
entity["organization","FinOps Foundation","finops org"] w 2025–2026 formalizuje to językiem „Scopes” (Cloud+ podejście) i rozszerza FinOps o SaaS i data center jako równorzędne zakresy kosztowe, co wprost pasuje do AI jako kolejnego scope’u. [6]
Specyfikacja FOCUS mówi o normalizacji danych kosztowych „cross-vendor” i jawnie wskazuje SaaS jako klasę generatorów danych billingowych. [7]

Konfrontacja z repo: masz świetną telemetrię i gating dla *supply chain* (SBOM/scan/delta/gate), ale nie widać analogicznej telemetrii dla AI-runtime (tokeny/czas/koszt/tenant/workflow) – czyli dokładnie tam, gdzie Bain i FinOps lokują główne tarcie rynkowe.

Agentowość i „inference whales”: ryzyko ekonomiczne = ryzyko overload

entity["organization","Business Insider","digital news outlet"] opisał zjawisko „inference whales”: ciężcy użytkownicy potrafią generować koszty inferencji rzędu dziesiątek tysięcy dolarów przy abonamencie setek dolarów, co wymusza limity i caps w modelach „unlimited”. [8]
To łączy ekonomię z niezawodnością: „whale” jest jednocześnie kosztem i obciążeniem.

Konfrontacja z repo: Twoje `swarm` ma właściwy odruch (rate limit, circuit breaker), a nawet wybór fail-closed (który chroni przed runaway cost, gdy rate-limit service nie działa). [9]
Ale ryzyko powrotne jest w samym sercu aplikacji (`aggregator.py`): brak backpressure jest punktem, gdzie agentowość (w AI) zamienia się w „bursty” i kaskady. [3]

Energia jako twarde ograniczenie skalowania AI

entity["organization","International Energy Agency","intergovernmental energy org"] prognozuje wzrost zużycia energii przez data centers do ok. 945 TWh w 2030 (base case), z AI jako istotnym driverem oraz z ryzykiem bottlenecków w sieci/planowaniu. [10]

Konfrontacja z repo: HMK9D posiada pojęcie „energii" (E(Δ), E_total) i „energy_guard", czyli masz konceptualny język, by w AI-SaaS mierzyć „energię" decyzji (koszt kroku).
Brakuje jednak spięcia z realnymi pomiarami (tokeny, latency, koszt GPU, energia/CO_2 jako metryka). To oznacza, że Twoja asymetria jest obecnie silniejsza jako **architektura sterowania** niż jako **architektura rachunku energii**.

Bezpieczeństwo LLM: OWASP jako „checklista ataków"

entity["organization","OWASP Foundation","security nonprofit"] Top 10 dla aplikacji LLM (v1.1) klasyfikuje ryzyka: prompt injection, insecure output handling, poisoning, model DoS, supply chain, disclosure itd. [11]

Konfrontacja z repo: jesteś mocny w dwóch obszarach OWASP: supply chain (SBOM) oraz część model DoS (rate limiting). [12]
Najsłabszy obszar (z perspektywy OWASP) to **insecure output handling** i „agency": brak twardych bramek semantycznych na wyjściu (np. walidacja schematem, allow-list narzędzi, sandbox wykonania). [13]

Governance i compliance: NIST + AI Act o monitoringu w cyklu życia

entity["organization","NIST","us standards institute"] AI RMF 1.0 i GenAI Profile kładą nacisk na funkcje Govern/Map/Measure/Manage, w tym testowanie przed wdrożeniem i regularnie „w operacji", monitoring ryzyk, mechanizmy override/appeal i plan monitoringu post-deployment. [14]
AI Act (PL, EUR-Lex) wymaga m.in. automatycznego rejestrowania zdarzeń dla systemów wysokiego ryzyka oraz post-market monitoringu (art. 72), z planem jako częścią dokumentacji technicznej, i z możliwą analizą interakcji z innymi systemami. [15]

Konfrontacja z repo: posiadasz budulce (CMM z hashami; event-sourcing w SBOM), ale brakuje „artefaktu spinającego": formalnego planu monitoringu (metryki, progi, właściciele, eskalacje, mechanizmy incident response). To jest dokładnie luka, którą AI Act i NIST próbują zamknąć na poziomie procesu, nie tylko technologii. [16]

„Disruption" na rynku software: sygnał Reuters/rynek kredytu

W lutym 2026 rynek potraktował agentową automatyzację jako realne ryzyko dla modeli „visibility premium" w software. W relacji o ryzyku na rynku pożyczek korporacyjnych entity["company","Morgan Stanley","investment bank"] wskazuje, że obawy o AI-disruption zaczęły przenikać do kredytu (software jako istotna część rynku loan), z koncentracją ryzyka w niższych ratingach. [17]
W materiale o „selloff" po toolingu entity["company","Anthropic","ai company, claude"] pojawia się teza o erozji „visibility premium" i ryzyku dla modelu per-user/per-seat, bo AI pozwala robić więcej „z mniejszą liczbą ludzi". [18]

Konfrontacja z repo: Twoje repozytoria są „anty-SaaS-owe" w tym sensie, że inwestują w sterowanie złożonością, a nie w maksymalizację skalowania „seat". To jest zgodne z kierunkiem krytyki. Jednocześnie, żeby falsyfikować narrację rynku (czyli pokazać, że da się utrzymać stabilność

ekonomiczną), potrzebujesz instrumentacji telemetrii AI-zużycia i kosztu w runtime – bo to jest dziś rdzeń pricing debate. [19]

Tabela porównawcza: fragmenty kodu vs. oceny mediów/ekspertów

Repo / fragment	Mechanizm asymetrii (co robi)	Ryzyko zewnętrzne, do którego trafia	Zgodność z krytyką	Luka drugi
`sbom/lab/jenkins/pipeline_one.pipeline`	SBOM+scan+deltas+`FAIL_ON` → stop wydania	Supply chain i „gating zamiast raportu”	Wysoka (twarde progi)	Nie o AI-ru (mod dane)
`sbom/kryptologia-informacyjna-sbom.md`	Koncepcja: pomiar→próg→akcja, „dowód pochodzenia”	Governance/ sterowanie ryzykiem	Wysoka (zgodna z NIST „Measure/ Manage”)	Wyma wdroż podp attest end-t
`swarm/.../rate-limit.yaml`	`failure_mode_deny: true` (fail-closed)	Model DoS + runaway cost	Wysoka; OWASP LLM04, LLM05	Rate- staje eleme kryty
`swarm/.../circuit-breaker.yaml`	outlier detection (ejection)	Kaskady awarii/overload	Wysoka; SRE load shedding/ cascades	Agres gener positi
`swarm/aggregator/aggregator.py`	wątek per pakiet, brak backpressure/timeout	Źródło kaskad wewnątrz usługi	Niezgodność (osłabia mesh-gating)	Ryzyk koszt środk
`HA2D/context_protocol.md`	pamięć kontekstu z `sha256`	Audyt/trace i integralność	Zgodność (budulec pod AI Act)	Brak plane monit metry
`chunk-chunk/hmk9d_protocol.yaml`	H(s)=g(F(s)), R(F,g), E(Δ), bramki	Agentowość, energia, progi	Bardzo zgodne kierunkowo	Brak opera telem
`ai_platform/platform.md`	/QV9D mapuje architekturę semantyczną na kod	Redukcja „mnożenia logik”	Zgodne z potrzebą governance	Brak deter ID (ry spójn
`writeups/...facebook_case.md`	formalna falsyfikacja protokołu (acc > baseline)	Evidence-based AI safety	Silne metodologicznie	Wyma walid out-o

Dla powiązań OWASP/NIST/AI Act/FinOps/Envoy/Istio/SRE zob. źródła: [20]

Statystyczna falsyfikacja hipotez asymetrii

Poniżej definiuję hipotezy w stylu „da się obalić” oraz minimalny zestaw metryk/testów. Jeśli brak danych z produkcji, doprecyzowuję eksperymenty i pokazuję symulację syntetyczną (ilustracyjną, nie dowodową).

Definicja operacyjna asymetrii

Asymetria = istotna redukcja ogonowego ryzyka (kosztowego, awaryjnego, bezpieczeństwa, compliance) przy relatywnie małym koszcie wdrożenia i przy zachowaniu funkcjonalności. W praktyce mierzymy ją jako zestaw efektów: - spadek prawdopodobieństwa incydentu (lub straty) w ogonie rozkładu, - spadek „CVaR” kosztu/awarii (średnia w najgorszych X% przypadków), - skrócenie czasu reakcji (MTTR / czas do detekcji / czas do stop-wydania), - poprawę auditability (kompletność logów/planów).

Hipotezy do falsyfikacji

H0-E (ekonomia): wprowadzenie budżetów użycia AI i bramek (rate limit / budżet tokenów / degrade) **nie zmienia** prawdopodobieństwa ujemnej marży per tenant i nie zmienia ogonów kosztu.
H1-E: prawdopodobieństwo straty i ogon kosztu spadają istotnie.

H0-R (reliability): mesh-gating (rate limit + circuit breaker) **nie zmniejsza** ryzyka kaskad (skok 5xx, retry amplification, p99 latency) w testach przeciążeniowych.
H1-R: zmniejsza istotnie.

H0-S (supply chain): SBOM-gating **nie redukuje** ekspozycji na Critical/High w wydaniach produkcyjnych.
H1-S: redukuje.

H0-G (governance): warstwa logów/pamięci kontekstu (CMM) **nie zwiększa** zdolności spełnienia wymogów monitoringu (AI Act/NIST) w sensie mierzalnym (kompletność logów, procedury, czas reakcji).
H1-G: zwiększa.

Metryki, dane wejściowe i testy

Hipoteza	Metryki (przykłady)	Dane wejściowe	Test(y) statystyczne
E	`P(loss)`; p95/p99/p999 kosztu; CVaR_5%; koszt/rezultat	logi tokenów, czas inferencji, koszty, billing, tenant	test dwóch proporcji (loss); bootstrap różnic kwantyli; Mann-Whitney/KS dla rozkładów
R	p99 latency; 5xx rate; liczba ejection; retry rate; error budget burn	load test + telemetria mesh; trace’y	porównanie powtórzeń A/B (paired); bootstrap; analiza szeregów czasowych
S	#Critical/#High na release; „dwell time” podatności; % wyjątków	eventy `scan/gate` z CI oraz repo CVE match	testy Poissona / regresja; różnica średnich; survival dla dwell time

Hipoteza	Metryki (przykłady)	Dane wejściowe	Test(y) statystyczne
G	kompletność rejestrów zdarzeń; czas wykrycia incydentu; audytowalność zmian	logi, CMM, runbooki, raporty	scoring + testy zgodności; analiza przed/po; audyt ścieżek

Źródła normatywne dla warstwy G: NIST AI RMF/GenAI Profile oraz AI Act (w tym art. 72 i wymogi rejestrowania zdarzeń). [21]

Symulacja syntetyczna „inference whales" jako test E

Ponieważ często brakuje danych produkcyjnych na starcie, można wykonać **symulację heavy-tail** na rozkładach obciążeń (to jest demonstracja mechanizmu, nie dowód o Twojej bazie klientów).

Założenia (syntetyczne, ale zgodne z obserwacją, że rozkłady użycia są „ogonowe" i że modele „unlimited" łapią whales): [22]
- 50k tenantów; liczba zadań/msc ~ Poisson(100)
- tokeny/zadanie ~ lognormal (średnia ok. 2000 tokenów, wysoka wariancja)
- koszt 0.6$ / 1k tokenów; abonament 200$

Wynik (z jednej symulacji Monte-Carlo):
- Bez budżetu: **~2.05% tenantów** generuje koszt > 200$ (ujemna marża); p99 koszt ≈ 223$.
- Z budżetem ≈ 333k tokenów/msc (próg break-even): **0% tenantów** przekracza koszt 200$; p99 koszt ≈ 200$; dotkniętych budżetem jest ~2% tenantów (powyżej progu).

Interpretacja falsyfikacyjna: jeśli na realnych danych *po wdrożeniu budżetów* nie obserwujesz spadku `P(loss)` ani ogonów kosztu, H1-E upada. Jeśli obserwujesz spadek, masz empiryczny argument, że Twoja architektura „bramek" daje przewagę ekonomiczną właśnie w ogonie (tam, gdzie rynek dziś cierpi). [22]

Luki, niezgodności i ryzyka

Największe luki są spójne z tym, co rynek i standardy uważają dziś za rdzeń problemu SaaS+AI.

Brak instrumentacji ekonomiki AI w runtime

Bain i FinOps mówią, że bez telemetrii nie da się przenieść pricingu na usage/outcome. [23]
Twoje repozytoria mają świetny wzorzec telemetrii dla SBOM, ale nie mają analogicznej, ustandaryzowanej telemetrii dla AI: tokeny, koszt, budżety, „unit cost per outcome". To jest luka, która może sprawić, że asymetria będzie realna technicznie, ale nieweryfikowalna ekonomicznie.

„Bramka na brzegu" vs „brak bramki w środku"

Mesh-policies są zgodne z praktykami przeciążeniowymi SRE, ale `aggregator.py` jest potencjalnym generatorem przeciążenia. SRE wprost ostrzega przed retry amplification i potrzebą load shedding/backoff. [3]
To jest ryzyko krytyczne, bo AI-agentowość w praktyce tworzy analogiczne wzorce (wiele kroków, retry, intensywny tool-use).

OWASP-LLM: braki w output-safety i „excessive agency”

SBOM i rate limiting adresują supply chain i DoS, ale OWASP wymienia też prompt injection i insecure output handling jako fundamentalne klasy ataków. [11]
W kodzie nie widać twardych „bramek semantycznych” (walidacja wyników, sandbox, polityki narzędzi). To oznacza, że przy dołożeniu warstwy LLM do istniejącej infrastruktury ryzyko przesunie się z „transportu” do „semantyki”.

AI Act: brakuje planu monitoringu jako artefaktu sterowania

AI Act (art. 72) wymaga post-market monitoring system i planu, a w PL wersji dodatkowo widać nacisk na rejestrowanie zdarzeń w cyklu życia (art. 12) jako wsparcie monitoringu. [24]
CMM i event-sourcing są dobrym budulcem, ale bez spięcia w plan (metryki/progi/właściciele/eskalacje) pozostaje ryzyko „compliance-by-logs”, czyli logi istnieją, ale nie sterują. [25]

QV9D: ryzyko spójności bez deterministycznych ID

`ai_platform` jawnie sygnalizuje brak deterministycznego sposobu wyliczania `id_latarni`. To ryzyko, bo bez stabilnych identyfikatorów nie zrobisz w pełni audytowalnego mapowania i korelacji metryk w czasie.

Rekomendacje techniczne i komunikacyjne

Rekomendacje są tak dobrane, aby działały przy „braku ograniczeń” budżetowych, ale każda ma też wersję minimalną (MVP) – w treści wskazuję, co jest rdzeniem.

Rekomendacje techniczne

Pierwszym ruchem powinno być „skopiowanie wzorca SBOM” na AI-runtime.

1) **AI-telemetria i bramki ekonomiczne (MVP w 2–4 tyg.)**
Zdefiniuj eventy analogiczne do `sbom/scan/delta/gate`, ale dla AI: `ai_usage_snapshot`, `ai_cost`, `ai_delta`, `ai_budget_gate`. Minimalny payload: tenant/workflow, tokeny, czas, model, narzędzia, koszt. Następnie ustaw progi budżetowe i polityki degradacji (np. tańszy model, limit iteracji, deny). To jest bezpośrednia odpowiedź na Bain/FinOps (telemetria → pricing) oraz na whale-problem. [26]

2) **Wpięcie kosztów w standard: FOCUS jako docelowy schemat**
Jeżeli dostawca (lub Twoja platforma) generuje billing, FOCUS jest najbliższym „językiem wspólnym” do normalizacji kosztów i usage w wielu scope’ach (w tym SaaS). To upraszcza analitykę i wzmacnia argumenty pricingowe. [27]

3) **Naprawa „wewnętrznego DoS”: backpressure, kolejki, timeouts**
Zastąp thread-per-packet kolejką + workerami z limitem, dodaj timeouts do HTTP, retry z exponential backoff, i mechanizm dropu (shedding). To jest dokładnie praktyka SRE przy przeciążeniu. [3]

4) **OWASP-LLM guardrails: wejście/wyjście/narzędzia**
Wprowadź: walidację outputu (schemat), allow-listę narzędzi, sandbox wykonania, limity iteracji agentów, ochronę przed prompt injection („system prompt isolation”, kontekst rozdzielony). OWASP Top 10 daje tu minimalną listę klas ryzyk. [11]

5) **Plan post-market monitoring jako plik repozytoryjny**
Zbuduj artefakt w repo (np. `monitoring_plan.md` + `metrics.yaml`), który mapuje: (a) metryki ryzyka, (b) progi, (c) właścicieli, (d) eskalacje, (e) retencję danych. Wprost nawiąż do AI Act art. 72 i NIST AI RMF (Measure/Manage). [28]

6) **Deterministyczne ID dla QV9D**
Zaprojektuj `id_latarni = hash(repo + path + rola + warstwa + most)`, z wersjonowaniem schematu. Wtedy QV9D staje się korelowalny w czasie i nadaje się do statystyki (ćwiczenie do falsyfikacji).

Rekomendacje komunikacyjne

1) **Komunikuj „sterowanie" metrykami, nie metaforami**
Rynek reaguje dziś na brak widoczności i brak telemetrii (Bain) oraz na erozję „visibility premium" (doniesienia o repricingu po automatyzacji agentowej). [29]
Twoja narracja powinna brzmieć jak: „mamy progi, mamy liczniki, mamy tryb degradacji, mamy politykę wyjątków".

2) **Włącz kontekst polski i regulacyjny**
W szczególności: Data Act zmniejsza bariery zmiany dostawcy chmury i znosi opłaty za switching od 12 stycznia 2027 r. – komunikacyjnie to przesuwa ciężar z lock-inu cenowego na interoperacyjność i semantykę. [30]
To jest spójne z Twoim QV9D: „interoperacyjność semantyczna" jako przewaga.

Źródła priorytetowe i mapa cytowań

Poniżej źródła najbardziej nośne (pierwotne/oficjalne) i ich rola w raporcie:

- entity["organization","NIST","us standards institute"] AI RMF 1.0 + GenAI Profile: język Govern/Map/Measure/Manage, monitoring post-deployment. [14]
- AI Act (PL, EUR-Lex) + omówienie art. 72: obowiązki logowania i post-market monitoring plan. [15]
- entity["organization","OWASP Foundation","security nonprofit"] Top 10 LLM Apps v1.1: klasy ryzyk aplikacji LLM. [11]
- entity["organization","FinOps Foundation","finops org"] + FOCUS: normalizacja danych kosztowych (cloud/SaaS), „Scopes" jako odpowiedź na kosztową złożoność. [31]
- entity["organization","International Energy Agency","intergovernmental energy org"]: energia jako constraint skalowania AI. [10]
- entity["organization","Bain & Company","management consulting"]: telemetria jako warunek transformacji pricingu AI-SaaS. [5]
- SRE (Google): overload, load shedding, retry amplification → kaskady. [3]
- Polska perspektywa Data Act: entity["organization","Ministerstwo Cyfryzacji","warsaw, poland"] oraz źródła PARP/CRN. [30]
- Repo-synteza Twoich artefaktów i ich mapowania na ryzyka:

---

[1] [9] https://www.envoyproxy.io/docs/envoy/latest/api-v3/extensions/filters/network/ratelimit/v3/rate_limit.proto.html
https://www.envoyproxy.io/docs/envoy/latest/api-v3/extensions/filters/network/ratelimit/v3/rate_limit.proto.html

2 https://istio.io/latest/docs/tasks/traffic-management/circuit-breaking/
https://istio.io/latest/docs/tasks/traffic-management/circuit-breaking/

3 https://sre.google/sre-book/service-best-practices/
https://sre.google/sre-book/service-best-practices/

4 15 https://eur-lex.europa.eu/eli/reg/2024/1689/oj/pol
https://eur-lex.europa.eu/eli/reg/2024/1689/oj/pol

5 19 23 26 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/

6 31 https://www.finops.org/insights/2025-finops-framework/
https://www.finops.org/insights/2025-finops-framework/

7 27 https://focus.finops.org/what-is-focus/
https://focus.finops.org/what-is-focus/

8 22 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

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

11 12 13 20 https://owasp.org/www-project-top-10-for-large-language-model-applications/
https://owasp.org/www-project-top-10-for-large-language-model-applications/

14 21 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

16 24 28 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

17 https://www.investing.com/news/stock-market-news/ailed-software-selloff-may-pose-risk-for-15-trillion-us-credit-market-says-morgan-stanley-4497161
https://www.investing.com/news/stock-market-news/ailed-software-selloff-may-pose-risk-for-15-trillion-us-credit-market-says-morgan-stanley-4497161

18 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

25 https://airc.nist.gov/airmf-resources/airmf/5-sec-core/
https://airc.nist.gov/airmf-resources/airmf/5-sec-core/

30 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