ChatGPT

Asymetria Twoich repozytoriów wobec ryzyk SaaS+AI w warunkach „Armageddon AI”

Potrzeby informacyjne i metoda

Potrzeby informacyjne (3–6): - Jakie *konkretne mechanizmy w kodzie/dokumentacji* (gating, delta, pętle sterowania, odporność na kaskady) wytwarzają **asymetrię** względem klasycznego SaaS+AI. - Jak te mechanizmy mapują się na **najnowsze oceny ryzyk**: awarie i kaskady, koszty inference i repricing, bezpieczeństwo aplikacji LLM, governance/zgodność (AI Act), energia jako constraint. - Czy repozytoria zawierają **falsyfikowalne hipotezy** (metryki, progi, testy), czy tylko narrację. - Jakie są **luki i ryzyka drugiego rzędu** (np. fail-closed na brzegu, a brak backpressure w środku) oraz gdzie asymetria może „wyparować”. - Jak zaprojektować **statystyczną falsyfikację** (metryki, testy, dane, eksperymenty, symulacje Monte Carlo) z perspektywy „brak ograniczeń”. - Jak przełożyć wyniki na **rekomendacje techniczne i komunikacyjne** zgodne z oczekiwaniami regulatorów, FinOps i rynku.

Zakres i kolejność źródeł. Najpierw przejrzałem repozytoria wskazane przez Ciebie, używając włączonego konektora GitHub [1] (api_tool). Kluczowe fragmenty przedstawiam jako *dowody wprost* (cytowane fragmenty kodu/plików w raporcie). Następnie skonfrontowałem je z aktualnymi (2024–2026) źródłami pierwotnymi i wysokiej jakości: standardami i instytucjami (NIST, AI Act, OWASP, FinOps/FOCUS, IEA) oraz oceną rynkową (m.in. Reuters/Bain). [2]

Uwaga o źródłach medialnych PL. Odwołany artykuł PB.pl okazał się niedostępny dla narzędzia (blokada/płatna ściana), więc jego tezę triangluję przez równoległe depesze/relacje (zwłaszcza Reuters) opisujące to samo zjawisko „paniki AI” w sektorach pośrednictwa finansowego. [3]

Definicja robocza asymetrii. W raporcie asymetria oznacza: *mierzalną redukcję ogonowego ryzyka (kosztowego/awaryjnego/bezpieczeństwa/compliance) na jednostkę złożoności wdrożenia*, czyli przewagę „dźwigni sterowania” nad klasycznym podejściem SaaS (feature→wzrost→MRR). Źródła o mechanice kaskad używam jako „logiki dowodu”: pozytywne sprzężenia i progi (thresholds) są formalnym szkieletem, na którym testujemy Twoje bramki i pętle. [4]

Streszczenie wykonawcze

Twoje repozytoria wykazują spójną, rzadką na rynku oś projektową: **„pomiar → próg → akcja”** jako fundamentalny prymityw sterowania złożonością. W `sbom` jest to pętla DevSecOps (SBOM→scan→delta→gate→blokada wydania). W `swarm` jest to pętla odporności na przeciążenia (rate limiting fail-closed + outlier detection), a w `chunk-chunk` i `ai_platform` — próba uogólnienia tej cybernetyki na semantykę (9D), energię/koszt kroku i mapowanie architektury. To jest rdzeń „asymetrii”: zamiast akumulować złożoność funkcji jak SaaS, próbujesz ją *zamykać* w pętlach sterowania, które mają niską złożoność, a duży wpływ na ogony rozkładów (awarie, koszty, ryzyko regulacyjne). [5]

Konfrontacja z oceną ekspertów i mediów pokazuje jednak lukę kluczową dla „Armageddon AI”: **Twoje repozytoria są bardzo mocne w supply chain i edge-resilience, ale jeszcze niewystarczająco operacjonalizują ekonomię AI-runtime (telemetria tokenów/kosztów/budżetów) oraz bezpieczeństwo semantyczne aplikacji LLM (OWASP LLM01/02/06)**. Rynek i analitycy wskazują, że

właśnie te obszary wywołują repricing i „panikę” (SaaS i kolejne branże), a konsultanci wskazują brak telemetrii i infrastruktury billing/finance jako główną barierę przejścia z per-seat do hybrydowych modeli AI. [6]

W warstwie falsyfikacji masz mocny fundament (warunek `acc(Ĥ) > acc_bazowa` i rozdzielenie „dowodu warunkowego” od falsyfikacji), ale brakuje jeszcze pełnego mostu do danych produkcyjnych i standaryzacji kosztów/zużycia (FOCUS) oraz do AI Act (post-market monitoring plan). To nie obala Twojej tezy — raczej wyznacza, gdzie asymetria jest **już dowodowa**, a gdzie jest **wciąż programem badawczym**. [7]

Repozytoria jako dowody asymetrii

`sbom`: asymetria supply chain i „twarde bramki” w CI/CD

W `lab/jenkins/pipeline_one.pipeline` implementujesz pełną pętlę: - generacja SBOM (Syft) i skan podatności (Grype), - budowa migawki komponentów i porównanie delty z poprzednią migawką, - bramka `FAIL_ON` z decyzją `STOP/GO`, - emisja zdarzeń do analityki (Elastic), po czym *egzekucja decyzji* (exit 10).

Dowód wprost (fragmenty):

```
choice(name: 'FAIL_ON', choices: ['none','critical','high'], description:
'Gate threshold')
...
CRIT=$(jq '[... select(.=="Critical")] | length' "$SCAN_JSON")
HIGH=$(jq '[... select(.=="High")] | length' "$SCAN_JSON")
...
if [ "$FAIL_ON" = "critical" ] && [ "$CRIT" -gt 0 ]; then
DECISION="STOP"; ...; fi
...
if [ "$DECISION" = "STOP" ]; then ... exit 10; fi
```

To jest asymetryczne, bo: - koszt implementacji bramki jest mały, - a wpływ na ogon ryzyka (wypchnięcie “Critical/High” poza system produkcyjny) jest duży.

Ten kierunek jest zgodny z regulacjami i praktykami supply chain: NIST (w kontekście EO 14028) definiuje SBOM jako formalny zapis relacji łańcucha dostaw i podkreśla jego rolę w transparentności i szybszej remediacji. [8]
Wpisuje się też w podejście SSDF (secure SDLC jako wspólny język praktyk). [9]

W `kryptologia-informacyjna-sbom.md` explicite rozróżniasz „wiedzę” (SBOM jako artefakt epistemiczny) i „sterowanie” (pętla cybernetyczna), co jest ważnym elementem Twojej tezy o złożoności. Ten dokument idzie dalej niż standardowy opis SBOM, bo próbuje osadzić SBOM jako marker relacji i bazę do delt, a nie „dokument compliance”. Jednocześnie zawiera dane statystyczne o rynku OSS (np. „98% kodu zawiera OSS”), które wymagają w raporcie wsparcia zewnętrznym źródłem, jeśli mają być traktowane jako twarde tezy. [10]

`swarm`: asymetria odporności na kaskady, ale ryzyko „asymetrii pozornej”

Masz dwa silne mechanizmy na brzegu systemu:

(a) Rate limiting z trybem fail-closed.
W `infrastructure/istio/policies/rate-limit.yaml`:

```
name: envoy.filters.http.ratelimit
typed_config:
  domain: "aggregator-api"
  failure_mode_deny: true
  rate_limit_service:
    grpc_service:
      ...
    timeout: 0.25s
```

Z punktu widzenia kaskad i ekonomii AI jest to strategiczne: `failure_mode_deny: true` oznacza, że w przypadku błędu kontaktu z usługą limitującą ruch, Envoy nie przepuszcza requestów („fail closed”), co ogranicza scenariusz runaway load/cost. Takie zachowanie jest zgodne z dokumentacją Envoy (failure_mode_deny → deny przy błędzie rate-limit service). [11]
Istio pokazuje analogiczną konfigurację w zadaniu rate-limit. [12]

(b) Circuit breaking / outlier detection.
W `infrastructure/istio/policies/circuit-breaker.yaml`:

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

To bezpośrednio wpisuje się w praktykę przeciwdziałania *cascading failures* opisaną w SRE: kaskada to awaria rosnąca wskutek dodatniego sprzężenia (np. overload→spadek capacity→jeszcze większy overload). [13]

Ryzyko drugiego rzędu: w `aggregator/aggregator.py` wewnątrz aplikacji tworzysz wątek na każdy pakiet UDP i robisz HTTP POST 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 typowy generator kaskad: jeśli ruch rośnie, rośnie liczba wątków, rośnie presja na zasoby, a system wchodzi w dodatnie sprzężenie (overload). SRE podkreśla, że bez load shedding i kontroli kolejek

system łatwo wchodzi w snowball (crash-loop, eskalacja). [13]
W efekcie możesz mieć **bramkę na brzegu** (mesh), ale możliwość kaskady **w środku** (aplikacja), co redukuje realną asymetrię.

`chunk-chunk`: asymetria „sterowania semantyką" i kosztów kroku

`hmk9d_protocol.yaml` formalizuje układ: - zachowanie systemu jako złożenie kompresji i polityki: `H(s) = g(F(s))`, - ryzyko globalne: `R(F,g)` jako expected loss, - model „energii" kroku: `E(Δ)` i `E_total`, - **bramki** (np. energy_guard max 0.8) i „próg-przejście" jako commit.

To jest ważne, bo wyprowadza od razu warunki falsyfikacji (ryzyko mierzone jako expected loss; energia jako koszt) zamiast samego „języka metafor". Mechanicznie to dobrze pasuje do tego, co rynek nazywa dziś agentowością: więcej kroków, dłuższe łańcuchy, większa eksplozja tokenów i kosztów, czyli rośnie ogon rozkładu. [14]

Jednak u Ciebie „energia" jest jeszcze w większości konceptualna. W „Armageddon AI" różnica między konceptem a sterowaniem zaczyna się na telemetrii: tokeny/czas/koszt/tenant/workflow i decyzja (cap, degrade, deny). Bain wskazuje wprost, że przejście do modeli usage/outcome jest trudne m.in. dlatego, że wiele firm „nie ma telemetrii produktu i infrastruktury billing/finance" dla metryk AI. [15]

`ai_platform`: asymetria przez wspólny układ współrzędnych /QV9D, ale brak krytycznego elementu

`platform.md` buduje tezę „mnożeniu logik" kontrując mapą odwracalną: - logiczny wolumin `/QV9D`: `[warstwa]/[most]/[architektura]/[rodzaj_artefaktu]/[id_latarni]`, - mapowanie katalogów i artefaktów w jednolity rejestr, - obietnica, że metryki i raporty można agregować po osiach 9D (SEMANTYKA-ENERGIA, PRÓG-PRZEJŚCIE itd.).

To jest potencjalnie **asymetryczne organizacyjnie**: redukuje koszty koordynacji i „rozjazd ontologii" między repo, zespołami i metrykami — czyli jeden z rdzeni złożoności w SaaS+AI (różne warstwy, różne KPI, brak spójności). [16]

Jednocześnie dokument sam wskazuje lukę krytyczną dla sterowania i audytu: deterministyczne `id_latarni` jest „NIEZIDENTYFIKOWANE — DO ZAPROJEKTOWANIA". Bez stabilnego ID nie da się w pełni: - korelować zdarzeń w czasie, - budować statystycznej falsyfikacji per moduł/warstwa, - zapewnić audytowalności (w praktyce: „data lineage" metryk).

`HA2D`: asymetria audytu i ciągłości kontekstu (CMM)

`context_protocol.md` opisuje Context Memory Manager: - rekordy z `uuid`, `timestamp`, `payload`, `sha256`, - operacje STORE/RETRIEVE/GET_LATEST, - procedurę weryfikacji integralności.

Jest to dobry budulec pod wymogi logowania i trace'owalności. AI Act dla systemów wysokiego ryzyka wymaga, aby systemy technicznie umożliwiały automatyczne rejestrowanie zdarzeń (logs) w cyklu życia i aby logi wspierały m.in. post-market monitoring (art. 12 → art. 72). [17]
Problem: CMM nie jest jeszcze spięte w plan monitoringu wymagany przez art. 72 (system, plan, elementy, template) — to luka procesowa, która w 2026 staje się wymogiem „twardym" (template planu w implementing act z terminem). [18]

`writeups`: asymetria „dowodu” i falsyfikacji protokołu kontekstu

W `protokoly_kontekstu_chunk-chunk_facebook_case.md` masz rzadki element: formalny warunek „częściowego poznania” czarnej skrzynki protokołu decyzyjnego:

- budujesz zbiór danych `D={(M_t, A_{t+1})}`,
- konstruujesz aproksymację `Ĥ`,
- warunek przewagi: `acc(Ĥ) > acc_bazowa`,
- a następnie opisujesz typowe mechanizmy falsyfikacji: overfitting bez out-of-sample, dryf systemu, źle dobrany baseline, korelacje pozorne.

To jest istotne, bo w „Armageddon AI” rynek karze projekty, które nie potrafią wykazać świadectw (governance, ekonomika, bezpieczeństwo). Twoje writeups pokazują, że umiesz formułować tezy jako hipotezy testowalne, a nie jako narracje. [19]

Konfrontacja z najnowszymi ocenami ekspertów i mediów

Poniżej syntetyzuję „co dziś boli” SaaS+AI i zestawiam z Twoimi repozytoriami.

Ryzyko repricingu i „paniki AI” jako kaskada rynkowa

W lutym 2026 media finansowe opisują kaskadowe przeceny „software” i kolejne sektory (wealth management/brokerzy, insurance brokers, porównywarki), po pojawieniu się narzędzi AI automatyzujących zadania dotąd zmonetyzowane przez pośredników. [20]
Mechanika jest zgodna z teorią kaskad progowych: mały impuls (nowa zdolność automatyzacji) zmienia progi decyzyjne uczestników rynku (inwestorów/klientów), uruchamiając szeroką falę (global cascade). [19]

Zgodność z Twoją logiką: Twoje repozytoria są kierunkowo „anty-SaaS-owe”: asymetria nie jest w funkcji UI, tylko w sterowaniu złożonością (gating, delta, obserwowalność). To jest zgodne z tezą rynku: jeśli AI „wchłania funkcje”, przewagę mają te systemy, które potrafią kontrolować ogony (koszty/ryzyko/compliance) i dostarczać dowody sterowania.

Koszty inference i „inference whales” jako ryzyko ekonomiczne i awaryjne

Business Insider opisuje zjawisko „inference whales”: użytkownicy/tenanci potrafią generować koszty w dziesiątkach tysięcy USD przy abonamencie rzędu 200 USD, co wymusza caps i redesign pricingu. [14]
To jest jednocześnie: - ryzyko marży (unit economics), - ryzyko niezawodności (runaway load i service disruption), czyli klasyczny przypadek wspólnego źródła kaskad: dodatnie sprzężenie w systemie kosztowo-obliczeniowym.

Zgodność z repo: masz mocną infrastrukturę do *gatingu* (fail-closed rate limit, dodatnie pętle opisane w HMK9D), ale nie widać jeszcze twardej instrumentacji kosztów AI-runtime i budżetów per tenant (co Bain wskazuje jako warunek przejścia do pricingu usage/outcome). [21]

Bezpieczeństwo aplikacji LLM: OWASP oraz przesunięcie ryzyk z „transportu” do „semantyki”

OWASP Top 10 dla aplikacji LLM wskazuje m.in. prompt injection (LLM01), insecure output handling (LLM02), model denial of service (LLM04) oraz supply chain (LLM05). [22]

Twoje repozytoria adresują: - LLM05 (supply chain) silnie przez `sbom`, - część LLM04 (DoS/koszty) przez rate limiting i circuit breaker w `swarm`.

Natomiast w kodzie nie widać jeszcze pełnego zestawu „guardrails" na wyjściu i w narzędziach (LLM02) ani polityk przeciw prompt injection i data exfiltration (LLM01/06). [22]
W świecie agentowym to właśnie output-handling (np. wykonywanie komend, zapisy do systemów, wywołania narzędzi) bywa najgroźniejszym kanałem kaskady: model generuje „akcję", która uruchamia lawinę zmian. (To spina się bezpośrednio z Twoim pojęciem „Próg–Przejście".)

Governance i compliance: NIST + AI Act jako formalizacja „pętli po wdrożeniu"

NIST AI RMF 1.0 i profil dla GenAI są ujęciem cyklu życia ryzyka (govern/map/measure/manage) i nacisku na pomiar oraz zarządzanie ryzykiem w operacji. [23]
AI Act (art. 12) wymaga automatycznego logowania zdarzeń przez systemy wysokiego ryzyka i wskazuje, że logi mają wspierać post-market monitoring (art. 72). [17]
AI Act art. 72 wymaga też formalnego **post-market monitoring system** i **post-market monitoring plan**, a Komisja ma wydać template planu do 2 lutego 2026. [24]

Twoje repo ma budulec (CMM, eventy w pipeline), ale brakuje jeszcze „artefaktu planu" (metryki, progi, właściciele, eskalacje, retencja, analiza interakcji systemów), czyli tego, co AI Act próbuje wymusić jako standard sterowania po wdrożeniu. [25]

Energia jako constraint: IEA i „Semantyka–Energia"

Raport IEA wskazuje, że zużycie energii centrów danych (ok. 415 TWh w 2024) ma według scenariusza bazowego wzrosnąć do ok. 945 TWh w 2030, z AI jako kluczowym driverem i z ryzykiem bottlenecków infrastrukturalnych. [26]
Twoje ujęcie energii (E(Δ), E_total, energy_guard) jest w tym kontekście trafne: jeśli energia staje się twardym ograniczeniem, „koszt kroku" musi być częścią protokołu. Luka znów dotyczy operacjonalizacji: bez telemetrii kosztu/energii w runtime, energy_guard pozostaje deklaracją.

FinOps/FOCUS i „telemetria jako warunek przetrwania w AI-SaaS"

FinOps Foundation wprowadza „Scopes" (Public Cloud, SaaS, Data Center) jako formalny sposób zarządzania różnymi segmentami kosztów; to istotne, bo AI-SaaS jest hybrydą naraz w kilku scope'ach. [16]
FOCUS 1.3 (ratyfikacja w grudniu 2025) normalizuje zbiory danych koszt/usage, dodając m.in. lepszą obsługę commitments i metadanych kompletności/świeżości. [27]
Bain podkreśla, że zmiana pricingu (seat→hybrid/usage/outcome) wymaga telemetrii oraz zdolności billing/finance i wspólnego języka między zespołami. [15]

Twoje repozytoria koncepcyjnie to przewidują (SEMANTYKA-ENERGIA, metryki jako koordynaty QV9D), ale nie pokazują jeszcze wprost „FOCUS-ready" datasetów i akcji (budżety, cap, degrade) jako systemu.

Tabela porównawcza: fragmenty repo vs. oceny ekspertów/mediów

Artefakt (repo)	Mechanizm asymetrii (co jest „dźwignią”)	Najnowsza ocena zewnętrzna (ryzyko)	Zgodność	Luka, która może skasować asymetrię	Jak test
`sbom` – Jenkins pipeline (SBOM→scan→delta→gate→exit 10)	Bramka wydaniowa oparta o severity + historia w analityce	Supply chain jako ryzyko systemowe; SBOM jako ułatwienie remediacji i transparentności; EO 14028/SSDF	Wysoka zgodność	Brak podpisów/ attestation w kodzie pipeline (dowód pochodzenia jako warunek zaufania)	`P(` prz dw
`swarm` – Envoy rate limit `failure_mode_deny: true`	Fail-closed chroni ogon (runaway load/cost)	LLM DoS i koszty (OWASP LLM04); „whales” i caps w branży	Zgodność kierunkowa	Rate-limit service staje się single point; brak budżetów per tenant w runtime	p99 test
`swarm` – outlier detection (circuit breaker)	Ejecting ogranicza dodatnie sprzężenia overload	SRE: kaskady jako pozytywne sprzężenie; potrzeba shed/ graceful degrade	Zgodność	Brak backpressure w aplikacji (thread per packet); ryzyko kaskady „wewnątrz”	Loa And late
`chunk-chunk` – HMK9D (H(s)=g(F(s)), R, E, gate)	Formalizacja tajl-ryzyka i kosztu kroku jako protokół	Rynek wymusza telemetrię i kontrolę kosztów agentów	Zgodność koncepcyjna	Brak mapowania E(Δ) na realne token/cost; brak wdrożonego budget-gate	`CV` po per
`ai_platform` – /QV9D mapowanie	Jedna ontologia metryk (redukcja mnożenia logik)	FinOps: scopes i wspólny język kosztów; AI Act: plan monitoringu	Zgodność	Brak deterministycznego `id_latarni` (traceability)	Cor ma me
`HA2D` – CMM (sha256)	Integralność pamięci i audyt zmian kontekstu	AI Act art. 12 logowanie + art. 72 monitoring plan	Częściowa	Brak formalnego planu monitoringu; brak wymogów minimalnych logów (np. czas użycia)	Aud sco

Artefakt (repo)	Mechanizm asymetrii (co jest „dźwignią”)	Najnowsza ocena zewnętrzna (ryzyko)	Zgodność	Luka, która może skasować asymetrię	Jak test
`writeups` – `acc(Ĥ)>acc_bazowa` + falsyfikacje	Nauka systemu jako hipoteza testowalna (out-of-sample)	NIST: pomiar i zarządzanie ryzykiem w cyklu życia	Wysoka metodologicznie	Bez danych i wersjonowania eksperymentu łatwo o overfitting narracji	Pre hol traf

Źródła wspierające wiersze tabeli: IEA energia/popyt, OWASP LLM Top 10, NIST AI RMF/GenAI Profile, AI Act art. 12 i 72, FOCUS/FinOps, Bain o telemetrii i pricingu, SRE o kaskadach, Reuters o repricingu i „panice AI”, Business Insider o „inference whales”. [28]

Czy kod i badania jasno wskazują problemy oraz czy falsyfikują krytykę

Gdzie repozytoria jasno „widzą” problem i zgadzają się z krytyką rynku: - **Kaskady/overload**: fail-closed rate limit + outlier detection to dokładnie rodzina mechanizmów rekomendowana do redukcji kaskad (SRE) i DoS/overload. [29] - **Supply chain**: SBOM jako sensor i bramka w CI/CD odpowiada trendowi regulacyjnemu i praktycznemu (EO/NIST/CISA) w kierunku „dowodu składu” i sterowalności. [30] - **Governance jako cykl życia**: CMM i pętle eventów są kompatybilne z AI Act logowanie/monitoring oraz NIST RMF (pomiar i zarządzanie w czasie). [31] - **Repricing w SaaS**: Twoja architektura zakłada, że przewaga jest w sterowaniu, nie w „funkcji, którą AI wchłonie” — to jest spójne z obserwacją rynkową o disruption po pojawieniu się narzędzi automatyzujących pracę wiedzy. [32]

Gdzie materiał jest falsyfikowalny (a więc naukowo mocny): - `sbom`: mierzalne outputy (liczba CVE, delta, decyzje gate, historia w analityce) → łatwy eksperyment przed/po. - `writeups`: jawny warunek `acc(Ĥ)>acc_bazowa` + katalog falsyfikacji (overfitting, drift, baseline) → poprawna epistemologia. - `swarm`: testowalne w load testach (p99 latency, 5xx, ejection) → falsyfikacja mechanizmu kaskad. [13]

Gdzie krytyka rynku jest trafna, a repozytoria jeszcze jej nie „domykają”: - **Telemetria AI i pricing**: Bain/FinOps/FOCUS wskazują telemetrię i zdolność billing jako warunek przejścia do stabilnych modeli pricingu. W repo masz język i konstrukcje (energia, metryki, mapy), ale brak „produkcyjnej bramki” na tokeny/koszty w runtime. [33] - **LLM output safety i agency**: OWASP LLM02/01/06 wymagają warstw walidacji i ograniczania działań. Repozytoria są bliżej „sterowania transportem” niż „sterowania semantyką outputu”. [22] - **AI Act plan monitoringu**: masz logowanie jako budulec, ale nie masz planu (wymaganego formalnie) jako artefaktu w repo. [18] - **Energia**: IEA pokazuje, że energia staje się twardym constraint; Twoje „E(Δ)” jest dobrze postawione, ale brak powiązania z realnymi danymi (token/CPU/GPU/CO_2). [26]

Wniosek: repozytoria są **zgodne z krytyką** w sferze diagnozy (kaskady, dowody, sterowanie), ale **nie falsyfikują jeszcze w pełni** krytyki ekonomicznej AI-SaaS (telemetria/pricing) i krytyki security-semantyki (OWASP LLM01/02/06), bo brakuje implementacji pomiaru i bramek w runtime.

Statystyczna falsyfikacja hipotez asymetrii i plan symulacji

Hipotezy (H0/H1) wprost falsyfikowalne

Poniżej proponuję minimalny zestaw hipotez, wprost spięty z Twoimi mechanizmami (bramki, pętle, delta). Wszystkie zakładają „brak ograniczeń" zasobowych, ale da się je realizować iteracyjnie.

H1-S (supply chain asymmetry).
- H0: po wdrożeniu SBOM-gate nie zmienia się istotnie odsetek wydań z `Critical`/`High`, ani czas ekspozycji (dwell time) dla CVE.
- H1: `P(release_with_critical)` spada, a dwell time maleje.

Dane: eventy z pipeline (`scan`, `gate`, `sbom_snapshot`, `delta`) + identyfikator wydania.
Testy: test dwóch proporcji (release with critical), regresja Poissona/neg-bin dla liczby CVE, survival analysis dla dwell time. Źródłowy sens SBOM jako narzędzia transparentności i remediacji — NIST/EO/CISA. [30]

H1-R (resilience asymmetry).
- H0: rate limit + outlier detection nie redukują ogonów opóźnienia i nie ograniczają kaskad przy przeciążeniu.
- H1: spadek `p99 latency`, `p99.9 latency`, `5xx rate`, wzrost stabilności w overload.

Dane: metryki Envoy/Istio + trace'y + logi aplikacji.
Testy: bootstrap różnic kwantyli; test KS dla rozkładów; analiza scenariuszy ramp i impulse (SRE rekomenduje oba). [13]

H1-E (economic asymmetry).
- H0: budżety AI-runtime nie redukują ogonów kosztu per tenant ani `P(loss)` (ujemnej marży) w modelu subskrypcyjnym/hybrydowym.
- H1: `CVaR95(cost_per_tenant)` i `P(loss)` spadają, przy kontrolowanym wpływie na NPS/retencję.

Dane: tokeny/time/model per tenant + koszt jednostkowy + revenue + churn.
Testy: różnica CVaR (bootstrap), test permutacyjny dla ogonów, regresja logistyczna dla `loss`. Kontekst rynkowy: „inference whales" i wymuszanie caps; telemetria jako bariera transformacji pricingu. [34]

H1-G (governance/compliance asymmetry).
- H0: dodanie logów/pamięci kontekstu nie zwiększa mierzalnie zdolności do spełnienia AI Act (logowanie, post-market monitoring).
- H1: rośnie kompletność logów i skraca się czas wykrycia/analizy incydentu; powstaje plan monitoringu spełniający template.

Dane: log completeness, coverage, MTTD/MTTR, audyty.
Testy: scoring zgodności (checklist) + analiza before/after; testy nieparametryczne dla MTTD/MTTR. Podstawa prawna: AI Act art. 12 (logs) i art. 72 (monitoring plan). [17]

Symulacja Monte Carlo jako demonstracja ogonów kosztu (gdy brak danych)

Poniższy wykres to **symulacja syntetyczna** (nie dowód o Twoich użytkownikach), pokazująca, dlaczego w AI-SaaS ogony kosztu są krytyczne, a prosta bramka „budżet per tenant" potrafi uciąć ekstremum:

- Założenia: heavy-tail tokenów (lognormal), 50k tenantów, koszt 0,60 USD / 1k tokenów, abonament 200 USD/mies. (przykładowo).
- Wynik: bez bramki maksymalny koszt tenantów przekracza abonament; z capem na break-even maks koszt jest ograniczony do 200 USD. Zjawisko „inference whales" jako realny driver zmian pricingu opisują media branżowe. [14]

Interpretacja w duchu Twojej teorii: **bramka** jest małym mechanizmem, który usuwa „dodatnie sprzężenie" kosztowe w ogonie (krok→kolejny krok→więcej tokenów→większy koszt). To jest asymetria. Zostaje jednak pytanie o koszt biznesowy (jakość/UX), który trzeba testować empirycznie.

Minimalne „eksperymenty zbierania danych" (żeby przejść od teorii do dowodu)

1) **Instrumentacja AI-runtime** (wprost pod H1-E): loguj tokeny, czas, model, narzędzia, tenant, workflow, wynik.
2) **FOCUS-ready dataset**: eksportuj cost/usage w formacie zbliżonym do FOCUS (albo mapowalnym), bo FOCUS wprost dąży do uniformizacji danych billing. [27]
3) **Testy przeciążeniowe**: ramp+impulse (SRE) na wejściu i wewnątrz aplikacji, bo kaskady często ujawniają się dopiero po przekroczeniu punktu krytycznego. [13]
4) **Audit trail** pod AI Act art. 12/72: logi „period of each use" i elementy post-market monitoring plan (art. 72). [17]

Luki, ryzyka i rekomendacje

Rejestr kluczowych luk i ryzyk

Ryzyko „Armageddon AI" w sensie ekonomicznym (repricing + whales): brak runtime-gating kosztu i telemetrii per tenant powoduje, że asymetria jest niepełna — rynek wprost obserwuje, że to wywołuje zmiany pricingu i panikę sektorową. [35]

Ryzyko kaskady in-system: `swarm` broni brzegu, ale `aggregator.py` może tworzyć dodatnie sprzężenie (wątki bez limitu, brak timeout, brak kolejki), czyli klasyczny generator cascading failures. [13]

Ryzyko OWASP LLM: brak warstw ochrony przed prompt injection i insecure output handling oznacza, że dodanie agentów może wytworzyć kaskady „semantyczne" (model wygeneruje akcję → akcja uruchomi lawinę). [22]

Ryzyko compliance: brak formalnego post-market monitoring plan (AI Act art. 72) i minimalnych logów (art. 12) może uczynić system nieaudytowalnym w reżimie wysokiego ryzyka. [25]

Ryzyko utraty traceability w QV9D: brak deterministycznego `id_latarni` utrudnia statystykę i dowody, a więc osłabia „asymetrię dowodową".

Rekomendacje techniczne (konkretne, egzekwowalne)

Wdroż runtime-telemetrię i bramki kosztowe, kopiując wzorzec `sbom` na AI.
- Zdarzenia analogiczne do `scan/gate/delta`, ale dla AI: `ai_usage_snapshot`, `ai_cost`, `ai_budget_gate`, `ai_degrade_action`.
- Budżet per tenant/workflow, z politykami: cap, degrade (tańszy model), deny (fail-closed) w krytycznych ścieżkach.
- Uzasadnienie: rynek już wymusza caps (inference whales), a Bain wskazuje telemetrię jako warunek transformacji pricingu. [34]

Ustandaryzuj cost/usage przez „FinOps scope thinking" i FOCUS.
- Traktuj AI jako osobny scope (obok cloud/SaaS/DC) i buduj eksport danych cost/usage zgodny lub mapowalny do FOCUS. [36]

Napraw „wewnętrzny generator kaskady" w `swarm`.
- Zamiast thread-per-packet: kolejka + worker pool + limit in-flight.
- W `requests.post`: timeout, retry z jitter i exponential backoff (SRE).
- Load shedding w warstwie aplikacji, nie tylko w mesh. [13]

Dodaj OWASP-LLM guardrails.
- Walidacja outputu schemą (structured output), allow-list narzędzi, sandbox, limity iteracji agentów, filtrowanie i klasyfikacja prompt-injection.
- Priorytet: LLM01/02/06 i LLM04. [22]

Domknij dowód SBOM kryptograficznie (attestation).
- SBOM jako dowód pochodzenia (podpis + powiązanie z digest artefaktu) jest spójny z kierunkiem EO 14028 i NIST. [8]

Zmaterializuj AI Act art. 72 jako plik w repo (plan monitoringu).
- Umieść `post_market_monitoring_plan.md` + `metrics.yaml`: metryki, progi, właściciel, eskalacja, retencja, analiza interakcji systemów.
- To bezpośrednio spełnia art. 72 i przygotowuje na template (termin 2 lutego 2026). [18]

Rekomendacje komunikacyjne (żeby asymetria była „czytelna dla rynku")

Komunikuj przewagę jako „sterowanie ogonem ryzyka", nie jako „więcej funkcji".
Rynek wycenia dziś ryzyko disruption i brak przewidywalności modeli biznesowych; przykładowo Reuters opisuje falę przecen i nerwowość wokół AI automatyzującej pracę usługową. [32]
Twoja narracja powinna więc pokazywać: - cap na koszty, CVaR, `P(loss)` (FinOps-język), - compliance-artefakty (AI Act), - kaskady i ich przeciwdziałanie (SRE + mesh + app).

Zestaw KPI do publicznej prezentacji (dowód asymetrii): - `CVaR95(cost_per_tenant)` i `P(loss)` (po wdrożeniu budżetów), - `p99 latency` + `5xx` w overload (przed/po), - `P(release_with_critical)` i dwell time CVE (przed/po), - „log completeness" i „monitoring plan coverage" (AI Act readiness). [37]

Diagramy zależności i kaskad

Pętla kosztowa AI-SaaS i miejsce na bramki (kaskada ekonomiczna):

```
flowchart LR
A[Więcej agentów / funkcji AI] --> B[Więcej kroków i tokenów]
B --> C[Wyższy koszt inference i obciążenie]
C --> D[Presja na pricing / caps]
D --> E[Spadek przewidywalności kosztów klienta]
E --> F[Churn / presja na marżę]
F --> A

C --> G[Bramka: budżet / degrade / deny]
G -->|ucina ogon| C
```

Twoja architektura „pomiar → próg → akcja" jako wspólny prymityw (SBOM + runtime + governance):

```
flowchart TB
subgraph SupplyChain["Supply chain (sbom)"]
SC1[SBOM + SCA scan] --> SC2[Delta] --> SC3[Gate FAIL_ON]
SC3 -->|STOP| SC4[Blokada release]
SC3 -->|GO| SC5[Publikacja artefaktu]
SC1 --> SC6[Eventy do analityki]
end

subgraph Runtime["Runtime (swarm + AI)"]
RT1[Ingress] --> RT2[Rate limit (fail-closed)]
RT2 --> RT3[Service]
RT3 --> RT4[Outlier detection / ejection]
RT3 --> RT5[Brak backpressure? ryzyko kaskady]
end

subgraph Governance["Governance (HA2D + AI Act)"]
GV1[Logi / CMM] --> GV2[Plan monitoringu]
GV2 --> GV3[Progi ryzyka i reakcje]
end

SC6 --> GV2
RT3 --> GV1
```

Priorytetowa lista źródeł (najcięższe „nośniki dowodu")

- National Institute of Standards and Technology [38] : AI RMF 1.0 i profil GenAI jako język cyklu życia ryzyka. [23]
- AI Act (art. 12 i 72 w serwisie interpretacyjnym UE) + polskie komunikaty rządowe o wejściu przepisów: logi i post-market monitoring plan. [39]
- OWASP Foundation [40] : OWASP Top 10 dla aplikacji LLM (LLM01–LLM10) jako mapa security-ryzyk. [22]
- FinOps Foundation [41] : Scopes (cloud/SaaS/DC) i FOCUS 1.3 jako standard danych cost/usage. [36]
- International Energy Agency [42] : energia jako constraint (415 TWh 2024 → ~945 TWh 2030). [26]

- Reuters [43] : bieżąca ocena rynkowa „AI disruption” (software selloff; rozszerzenie paniki na brokerów/wealth management). [44]
- entity["company","Bain & Company","management consulting firm"]: repricing i teza „telemetria jako warunek” przejścia do modeli hybrydowych/usage/outcome. [15]
- Business Insider o „inference whales” jako empiryczny symptom ogonów kosztu w AI-SaaS. [14]
- SRE Book (Google): definicja i mechanika cascading failures oraz praktyki load shedding. [13]
- Duncan J. Watts [45] i Mark Granovetter [46] : formalne modele progowe kaskad (szkielet teoretyczny „mały impuls → duża fala”). [47]

---

[1] [26] [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

[2] [23] 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

[3] https://www.reuters.com/business/us-brokerages-fall-ai-driven-rout-extends-financials-2026-02-10/
https://www.reuters.com/business/us-brokerages-fall-ai-driven-rout-extends-financials-2026-02-10/

[4] [5] [13] [29] [41] [45] https://sre.google/sre-book/addressing-cascading-failures/
https://sre.google/sre-book/addressing-cascading-failures/

[6] [20] [32] [44] https://www.reuters.com/business/retail-consumer/retail-inflows-into-software-stocks-hit-record-despite-ai-disruption-worries-2026-02-10/
https://www.reuters.com/business/retail-consumer/retail-inflows-into-software-stocks-hit-record-despite-ai-disruption-worries-2026-02-10/

[7] [27] https://focus.finops.org/focus-specification/
https://focus.finops.org/focus-specification/

[8] [10] [30] [43] https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-software-1
https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-software-1

[9] [40] https://csrc.nist.gov/pubs/sp/800/218/ipd
https://csrc.nist.gov/pubs/sp/800/218/ipd

[11] https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/rate_limit_filter
https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/rate_limit_filter

[12] https://istio.io/latest/docs/tasks/policy-enforcement/rate-limit/
https://istio.io/latest/docs/tasks/policy-enforcement/rate-limit/

[14] [34] [35] 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

[15] [21] [33] 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/

[16] [36] [37] https://www.finops.org/insights/2025-finops-framework/
https://www.finops.org/insights/2025-finops-framework/

[17] [31] [38] [39] [46] 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

[18] [24] [25] [42] 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

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

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

47 https://pmc.ncbi.nlm.nih.gov/articles/PMC122850/
https://pmc.ncbi.nlm.nih.gov/articles/PMC122850/