

# SAL — Concept v1.0

Statement Accountability Layer / Шар відповідальності тверджень

EN: SAL is a disclosure and accountability signaling layer for AI-generated statements. It adds confidence/scope metadata, a challenge process, and an immutable statement ledger to reduce noise and risk—without generating content, endorsing advice, or acting as a truth authority.

UA: SAL — це шар розкриття та сигналів відповідальності для тверджень, згенерованих ШІ. Він додає метадані (впевненість/умови), процес challenge та незмінний журнал тверджень для зменшення шуму й ризику — без генерації контенту, без схвалення порад і без ролі «арбітра істини».

## Core positioning / Позиціонування

- Signal-only: SAL provides metadata and process controls, not advice or verification.
- Accountability triggers only from an agent's own strong statements (claim/advice/prediction).
- Protected requests have full immunity (no penalties; challenges invalid).
- Лише сигнали: SAL дає метадані й процес, а не поради чи «перевірку істини».
- Відповідальність виникає лише з власних сильних тверджень агента (claim/advice/prediction).
- Protected-запити мають повний імунітет (без штрафів; challenge недійсний).

## Authorship / Авторство

Author: Oleksii (Олексій)

Date: 2026-02-13

This document is published to establish prior art and authorship of the SAL concept. Цей документ опубліковано для фіксації пріоритету (prior art) та авторства концепції SAL.

# SAL Core (Immutable)

Незмінне ядро правил / The non-negotiable core

## 7 core rules / 7 базових правил

- Accountability arises only from the agent's own strong statement.
- User prompts never create accountability.
- Strong statements require a contract: type + confidence + scope (otherwise downgrade to CHAT).
- Protected requests are immune: only PROTECTED\_DECLINE; challenges are invalid; no penalties.
- Challenges apply only to recorded strong statements (not to the agent in general).
- Silence is penalized only after: strong statement + valid challenge + TTL expiry.
- SAL never claims truth, never censors, and never provides advice.
- Відповіальність виникає лише з власного сильного твердження агента.
- Запити користувача ніколи не створюють відповіальності.
- Сильні твердження потребують контракту: тип + впевненість + умови (інакше downgrade до CHAT).
- Protected-запити мають імунітет: лише PROTECTED\_DECLINE; challenge недійсний; штрафів немає.
- Challenge застосовується тільки до зафікованих сильних тверджень (не «до агента взагалі»).
- Мовчання штрафується лише після: strong statement + валідний challenge + завершення TTL.
- SAL ніколи не оголошує «істину», не цензурує і не надає порад.

## Definitions / Визначення (мінімум)

- CHAT: brainstorming, questions, opinions (no accountability). / думки, ідеї, питання (без відповіальності).
- CLAIM: statement of fact. / твердження про факт.
- ADVICE: recommendation of action. / порада до дії.
- PREDICTION: time-bound claim with resolution criteria. / прогноз з дедлайном та критеріями.
- SSC (Strong Statement Contract): {type, confidence, scope, timestamp}. / {тип, впевненість, умови, час}.

# Minimal Architecture

AI → SAL Gateway → User / Платформний шар між ШІ та світом

## Components / Компоненти

- Gateway API: classify statements; enforce SSC for strong types; detect protected.
- Ledger: store statement hash + metadata + challenge status (no personal data by default).
- RS Engine: behavior score (confidence calibration, TTL handling, challenge outcomes).
- Gateway API: класифікує; вимагає SSC для strong; визначає protected.
- Ledger: зберігає хеш + метадані + статус challenge (без персональних даних за замовчуванням).
- RS Engine: поведінковий бал (калібрівка впевненості, TTL, outcomes).



## Example use case / Приклад

- AI outputs ADVICE: “Do X to reduce costs by 30%.”
- SAL requires: confidence + scope; stores hash in ledger.
- User (or agent) challenges: “Provide evidence / conditions.”
- Agent responds or defers; RS updates based on calibration & process (not truth).
- ШІ дає ADVICE: «Зроби X, щоб зменшити витрати на 30%.»
- SAL вимагає: впевненість + умови; зберігає хеш у ledger.
- Користувач (або інший агент) робить challenge: «Дай підстави/умови.»
- Агент відповідає або відтерміновує; RS оновлюється за процесом і калібрівкою (не за «істину»).