



SWISSI INSTITUTE FOR AI

Formal Multi-Agent AI System Architecture

Generic AI Framework Development under Solvency II and AI Act in Austria and Germany

Walter Kurz¹

¹Swissi Institute for AI, office@swissi-ai.institute

May 2025

Abstract

This paper proposes a formal multi-agent architecture for implementing enterprise AI in regulated insurance firms, integrating economic theory with institutional design. The framework synthesises three core theoretical perspectives: Arrow’s risk pooling theory to formalise risk transformation under uncertainty, Nash equilibrium to model strategic interactions between decision agents, and Principal-Agent theory to address incentive alignment under information asymmetry. The insurer is modelled as a constrained optimisation entity operating under solvency, legal, ESG, and operational boundaries, with specific focus on the regulatory contexts of Austria and Germany. The architecture decomposes the firm into multiple specialised agents—each representing distinct functional domains such as capital management, underwriting, claims processing, compliance, fraud detection, and client interaction. Human-in-the-loop agents are integrated through a tiered access control system, ensuring differentiated data visibility and decision influence based on user roles. An orchestrator agent supervises inter-agent coordination, enforcing regulatory admissibility and institutional coherence under frameworks such as Solvency II, the AI Act, and the Insurance Distribution Directive. Protocol integration is based on asynchronous execution and dual-layer communication infrastructures, specifically the Model Context Protocol (MCP) and Agent-to-Agent (A2A) messaging. This structure enables the systematic design of compliant, auditable multi-agent systems aligned with the institutional logic of financial firms in Austria and Germany.

Keywords: Multi-Agent Systems, Enterprise AI, Insurance Firms, Solvency II, AI Act, Regulated Environments, Constrained Optimisation, Principal-Agent Theory, Nash Equilibrium, Arrow’s Risk Pooling, Austria, Germany, Institutional Design, ESG Compliance, Regulatory Architecture, Model Context Protocol (MCP), Agent-to-Agent Protocol (A2A), AI Governance, Algorithmic Accountability, Financial Regulation

1 Introduction

In this paper, we develop a theoretical framework to formalise the core institutional functions of insurance companies operating in Austria and Germany. Our objective is to represent these firms as systems of constrained optimisation problems, suitable for implementation via a decentralised multi-agent architecture. We deliberately avoid task-level AI applications and instead focus on the structural transformation of insurance firms into formal agent-based systems governed by economic objectives and regulatory constraints. The analysis is restricted to profit-oriented private insurers, enabling the derivation of objective functions consistent with utility or surplus maximisation under bounded rationality. This work contributes to a foundation for

applying advanced AI systems to insurance by grounding agent behaviour in firm theory rather than heuristic rule sets or isolated prediction models.

1.1 Institutional Scope: Insurance Markets in Austria and Germany

Austria and Germany constitute two of the most developed and stringently regulated insurance markets in Europe. Both jurisdictions operate under the Solvency II framework, characterised by high market penetration, standardised supervisory practices, and formally codified capital adequacy regimes. The German market includes a significant share of mutuals and public-law insurers, whereas Austria exhibits stronger concentration among private joint-stock firms. Regulatory supervision is exercised by BaFin in Germany and the FMA in Austria, under harmonised EU directives. These institutional similarities—amplified by coercive and normative isomorphic pressures within the EU regulatory field—permit a joint theoretical treatment of profit-driven insurer behaviour [1]. Capital requirements and solvency constraints are explicitly quantified under Solvency II, making it possible to model insurance firms as surplus-maximising entities operating under regulatory risk boundaries [2].

1.2 Taxonomy of Insurance Firms

Insurance firms in Austria and Germany can be categorised according to ownership structure, legal form, and regulatory mandate. Broadly, three types dominate the institutional landscape: (i) public-law insurers, such as statutory health insurance providers; (ii) mutual insurers, owned by their policyholders; and (iii) joint-stock companies, operating under shareholder control. While all entities are subject to Solvency II and supervised by national authorities, their internal objectives differ substantially. Public-law insurers are typically constrained by statutory duties and fixed benefit structures, with limited discretion over pricing or capital strategy. Mutuals operate under collective ownership, often prioritising member benefits over profit maximisation. In contrast, joint-stock insurers pursue surplus generation under market competition, subject to regulatory capital and risk constraints [3, 4]. These differences affect the formalisation of objective functions and admissible decision sets. Given this divergence, the subsequent analysis focuses exclusively on profit-driven private insurers, for which the firm can be modelled as a utility-maximising entity operating under solvency and compliance constraints.

1.3 Related Work

The theoretical foundation for modelling insurer behaviour rests on three principal frameworks. First, Arrow’s treatment of risk-bearing institutions provides a canonical basis for interpreting insurance as a mechanism for intertemporal risk transfer and welfare improvement under uncertainty [5]. Second, the principal-agent framework is widely used to model asymmetries between insurers and policyholders, particularly in underwriting and claims settlement, where hidden information and hidden actions affect contract outcomes [6, 7]. Third, the Nash equilibrium concept enables formal analysis of insurer interaction in competitive environments, such as premium setting, reinsurance strategy, or capital allocation, where strategic interdependence governs outcomes [8]. These foundations allow insurer behaviour to be framed as a set of constrained optimisation problems under uncertainty and regulation. This abstraction is essential for translating institutional logic into formal structures usable in multi-agent systems.

Beyond economic formalism, the internal structure and governance of insurance firms have been analysed through organisational theory. Principal-agent models extend to internal hierarchies, capturing incentive misalignments between shareholders, executives, and operational units [9]. Institutional isomorphism has been observed in the convergence of firm structures across markets with similar regulatory constraints, reinforcing the adoption of standardised roles and procedures [1]. This theoretical lens provides a basis for decomposing insurer operations into semi-autonomous subfunctions governed by compliance and incentive compatibility. Such decomposition aligns naturally with agent-based modelling approaches, in which decentralised decision entities operate under bounded rationality within a shared institutional environment.

Applications of AI in the insurance industry have focused primarily on operational subdomains such as fraud detection, customer profiling, and claims triage. Supervised learning techniques dominate, often relying on structured policyholder data or claim records [10]. More recently, natural language processing has been applied to extract and classify information from unstructured documents in claims workflows [11]. Industry-driven approaches increasingly promote intelligent automation platforms that combine machine learning with robotic process automation for end-to-end claims handling [12]. Such implementations remain

task-specific and do not extend to the systemic modelling of insurer behaviour. Existing AI solutions typically optimise local performance metrics without incorporating institutional objectives or regulatory constraints, limiting their suitability for architecture-level integration.

1.4 Contribution

This paper contributes a theoretical framework that formalises the core institutional functions of profit-driven insurance firms as constrained optimisation problems grounded in economic theory. Building on established models of utility, risk, and contract theory, we decompose insurer operations into analytically distinct subfunctions aligned with firm-level objectives. These subfunctions are then mapped onto a multi-agent system architecture, in which autonomous agents operate under bounded rationality, institutional constraints, and shared utility goals. Unlike task-oriented AI applications, the proposed structure enables systemic modelling of insurance firms as distributed decision systems, opening a path toward compliant, architecture-level AI implementation in regulated environments.

2 Formal Modelling of Insurance Firm Functions

The formal behaviour of an insurance company can be represented as an optimisation problem in which the firm seeks to maximise surplus or expected utility, subject to a set of internal and external constraints. The objective reflects the insurer’s economic role as an intermediary that transforms individual risks into collective stability through pooling and capital management. Constraints arise from capital requirements, risk boundaries, and regulatory compliance obligations, which delimit the admissible decision space. In regulated environments such as Austria and Germany, these constraints include solvency directives such as Solvency II, which formalise market-consistent valuation, capital adequacy, and risk-based supervision [13, 14], consumer protection rules under the Insurance Distribution Directive (IDD), which aim to enhance transparency, improve advisory standards, and strengthen product governance in insurance distribution [15, 16], and increasingly also algorithmic accountability derived from the AI Act, the GDPR, and ESG-related regulatory frameworks [17, 18]. This abstraction reduces the insurance firm to a constrained optimisation entity, forming the analytical basis for decomposing its internal functions into subcomponents represented by decision agents. As a baseline, the firm’s behaviour can be formalised as the following maximisation problem:

$$\max_{\mathbf{a} \in \mathcal{A}} \mathbb{E}[U(\pi(\mathbf{a}))] \quad (1)$$

$$\text{s.t. } \mathbb{E}[L(\mathbf{a})] \leq C \quad (\text{Capital adequacy constraint}) \quad (2)$$

$$\text{VaR}_\alpha(\pi(\mathbf{a})) \leq R \quad (\text{Risk exposure limit under Solvency II}) \quad (3)$$

$$\mathbf{a} \in \mathcal{D}_{\text{legal}} \quad (\text{Regulatory admissibility: AI Act, GDPR, IDD}) \quad (4)$$

$$\mathbf{a} \in \mathcal{E}_{\text{ESG,E}} \quad (\text{Environmental screening under EU Taxonomy and SFDR}) \quad (5)$$

$$\mathbf{a} \in \mathcal{G}_{\text{ESG,S\&G}} \quad (\text{Social and governance constraints}) \quad (6)$$

$$\mathbf{a} \in \mathcal{O}_{\text{operational}} \quad (\text{Operational feasibility}) \quad (7)$$

In this formulation, $\mathbf{a} \in \mathcal{A}$ denotes the vector of firm-level decisions, including pricing, underwriting, claims handling, investment allocation, and governance policies. The function $\pi(\mathbf{a})$ represents the surplus or profit generated by these decisions, while $U(\cdot)$ is the firm’s utility function, potentially reflecting risk-neutral or risk-averse preferences.

The capital adequacy constraint $\mathbb{E}[L(\mathbf{a})] \leq C$ ensures that expected liabilities do not exceed available capital C . Risk exposure is formalised via a Value-at-Risk condition at confidence level α , such that $\text{VaR}_\alpha(\pi(\mathbf{a})) \leq R$, where R is the firm’s Solvency II risk threshold. The admissible decision space is further bounded by legal and institutional requirements. The legal constraint $\mathbf{a} \in \mathcal{D}_{\text{legal}}$ enforces compliance with supervisory regimes such as the AI Act [17], the General Data Protection Regulation (GDPR), and the Insurance Distribution Directive (IDD) [15]. ESG admissibility is introduced via two distinct constraint sets: $\mathbf{a} \in \mathcal{E}_{\text{ESG,E}}$ denotes environmental eligibility under instruments such as the EU Taxonomy Regulation (EU) 2020/852 [19] and the Sustainable Finance Disclosure Regulation (SFDR) [20], including measurable thresholds like carbon intensity per revenue unit or share of taxonomy-aligned investments. $\mathbf{a} \in \mathcal{G}_{\text{ESG,S\&G}}$ ensures that actions meet minimum social and governance criteria, including metrics such as board diversity ratios, executive pay dispersion, transparency of grievance mechanisms, and policyholder participation in governance. $\mathbf{a} \in \mathcal{O}_{\text{operational}}$ guarantees that all selected actions remain within the firm’s technical and procedural capabilities.

This formulation expresses the insurer’s decision logic as a constrained optimisation problem: maximising expected utility while navigating capital limits, solvency thresholds, regulatory boundaries, and ESG accountability. It reflects the institutional reality that insurance firms are not profit-maximising entities in a vacuum, but regulated intermediaries embedded within a system of legal, financial, and sustainability norms. The constraint structure provides a formal foundation for balancing these competing obligations within an integrated decision architecture. The constrained optimisation problem can be reformulated using a Lagrangian representation, which incorporates the firm’s decision space and associated constraints into a single augmented objective:

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\lambda}) = \mathbb{E}[U(\pi(\mathbf{a}))] - \lambda_1 (\mathbb{E}[L(\mathbf{a})] - C) - \lambda_2 (\text{VaR}_\alpha(\pi(\mathbf{a})) - R) - \sum_{i=3}^6 \lambda_i \cdot \mathbb{I}_{\text{viol}}^{(i)}(\mathbf{a}) \quad (8)$$

In this formulation, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_6)$ denotes the vector of non-negative Lagrange multipliers associated with the respective constraints. The indicator functions $\mathbb{I}_{\text{viol}}^{(i)}(\mathbf{a})$ take the value 1 when the corresponding constraint is violated, and 0 otherwise. This representation allows each binding constraint to be interpreted as an implicit cost component: a marginal penalty or trade-off that reduces the firm’s attainable utility. The multipliers quantify the shadow price of each constraint — i.e. how much expected utility would improve if that constraint were marginally relaxed.

Put differently, this expression formalises the reality that insurers are not just maximising profit; they are doing so under legal, financial, and institutional boundaries, each of which carries a hidden opportunity cost. The formulation makes those trade-offs explicit and measurable.

This prepares the ground for decomposing the global optimisation problem into analytically tractable subcomponents, each corresponding to a distinct functional domain within the firm. These subcomponents will later be mapped to autonomous decision agents, coordinated within a multi-agent system aligned with the firm’s overall objective.

The formal constraint structure outlined above provides a general decision-theoretic foundation. To refine it, we now examine how different theoretical frameworks contribute distinct modelling perspectives to insurer behaviour. We begin with Arrow’s treatment of risk-bearing institutions. Arrow’s analytical framework interprets insurance as a mechanism for transferring individual uncertainty into collective stability through risk pooling under conditions of incomplete information and risk aversion [5].

Within this view, the firm functions as a utility-transforming intermediary, accepting idiosyncratic risks from clients and aggregating them into diversified portfolios whose outcomes are more predictable at the collective level. Risk-averse agents maximise the expected utility of their final wealth, implying that the insurance contract must improve the expected utility of the insured while maintaining the insurer’s solvency. The firm’s objective thus incorporates a concave utility function, and optimal contract design becomes a question of balancing marginal utility across risk classes subject to regulatory and capital constraints. This foundation legitimises the use of expected utility in the global model and provides a formal link between micro-level risk aversion and firm-level surplus transformation.

Following the expected utility framework established by Arrow [5], if W denotes the insurer’s terminal wealth and X the aggregate uncertain claim distribution, the firm solves:

$$\max \quad \mathbb{E}[U(W - X)] \quad (9)$$

where $U(\cdot)$ is a strictly concave utility function reflecting risk aversion, and X represents the net realised liabilities from pooled insured risks. This structure underpins the expected utility formulation applied in the global optimisation model.

A second theoretical perspective relevant to insurer modelling derives from principal–agent theory, which formalises the implications of asymmetric information between contracting parties. In insurance, such asymmetries are structural: the policyholder holds private information about risk type (adverse selection) and actions taken post-contract (moral hazard) [6]. This leads to inefficiencies in underwriting, pricing, and claims settlement, where the insurer must design mechanisms to extract truthful signals or induce appropriate behaviour.

Within the firm itself, principal–agent problems arise in governance structures, where shareholders (principals) must align the actions of executives and operational units (agents) under limited observability and incentive misalignment [9]. In the modelling context, principal–agent theory justifies the use of informational constraints in the firm’s decision space. These constraints restrict the feasible set of actions not only by regulation and capital, but by the need to maintain incentive compatibility.

The global optimisation problem is thus shaped by contracts and internal mechanisms that ensure delegated decisions align with firm objectives despite decentralised knowledge and interests. Following the principal-agent framework formalised by Rothschild and Stiglitz [6], an incentive-compatible mechanism can be expressed as:

$$a^* \in \arg \max_{a \in \mathcal{A}} \mathbb{E}[u(a, s(a)) \mid \theta] \quad (10)$$

where a^* is the equilibrium action, $u(\cdot)$ is the agent's utility, and θ is the agent's private type. This structure embeds information asymmetry into the admissible decision structure of the firm. Similarly, Nash equilibrium provides the formal language to describe strategic interactions among insurers in competitive markets, as outlined by Dickson and Drekić [8]. Each firm chooses a strategy that maximises its objective given the strategies of others.

This strategic context introduces equilibrium constraints into the firm's optimisation problem, particularly in areas where regulatory frameworks permit competition under capital requirements and fair disclosure. Premium rates, for example, are not solely determined by loss expectations, but also by competitive pressures and reactions. Similarly, capital allocation and reinsurance retention levels are shaped by the structure of rival firms' positions. In this setting, the insurer's feasible set is partially endogenous: constrained not only by internal and external norms, but by strategic stability.

Following the Nash equilibrium framework as applied to insurance markets by Dickson and Drekić [8], the equilibrium condition can be formalised as:

$$\mathbf{a}_i^* \in \arg \max_{\mathbf{a}_i \in \mathcal{A}_i} \mathbb{E}[U_i(\pi_i(\mathbf{a}_i, \mathbf{a}_{-i}))] \quad (11)$$

where each firm maximises its own expected utility conditional on the strategies of its competitors. This condition defines a Nash equilibrium in which no insurer has an incentive to deviate unilaterally from its chosen action.

The three modelling perspectives outlined above—Arrow's utility-based risk pooling [5], principal-agent theory under asymmetric information [6], and Nash equilibrium in strategic competition [8]—each contribute distinct structural constraints and behavioural mechanisms to the insurer's decision space. Together, they represent a layered interpretation of insurance firm behaviour: risk transformation, incentive design, and interdependent strategy selection.

To capture this multidimensional structure, we now introduce a unified formulation in which the firm is modelled as a distributed decision system composed of interdependent subfunctions. This approach allows each theoretical lens to be encoded as a constraint or objective component in a global architecture that reflects the institutional and economic realities of regulated insurance markets. It also provides a formal foundation for decomposing the firm into agent-level components in later sections. We define the unified model as follows:

$$\max_{\mathbf{a}_1, \dots, \mathbf{a}_n} \sum_{i=1}^n \mathbb{E}[U_i(\pi_i(\mathbf{a}_i, \mathbf{a}_{-i}))] \quad (12)$$

$$\text{s.t. } \mathbb{E}[L_i(\mathbf{a}_i)] \leq C_i \quad \forall i \quad (\text{Capital adequacy}) \quad (13)$$

$$\text{IC}_i(\mathbf{a}_i, \theta_i) \text{ holds } \forall i \quad (\text{Incentive compatibility}) \quad (14)$$

$$\mathbf{a}_i \in \mathcal{B}_i(\mathbf{a}_{-i}) \quad \forall i \quad (\text{Best response: strategic consistency}) \quad (15)$$

$$\mathbf{a}_i \in \mathcal{D}_i \cap \mathcal{E}_i \cap \mathcal{G}_i \cap \mathcal{O}_i \quad \forall i \quad (\text{Legal, ESG, operational admissibility}) \quad (16)$$

Here, the firm is composed of n subfunctions or decision units, each denoted by index i . Each $\mathbf{a}_i$ represents a vector of decisions associated with a particular function (e.g. pricing, underwriting, claims). The function $\pi_i(\mathbf{a}_i, \mathbf{a}_{-i})$ denotes the financial contribution of function i , which may depend on its own decisions as well as those of other units (interdependencies). $U_i(\cdot)$ is the local utility function, potentially reflecting risk preferences or performance targets. Constraint (13) ensures local capital adequacy; (14) imposes incentive compatibility under private information; (15) enforces strategic consistency in the presence of inter-agent competition or coordination; and (16) collects external admissibility constraints from regulatory, environmental, governance, and operational domains.

This structure treats an insurance company as a system made up of smaller expert units. Each unit has its own job and goals, but they all work together within a shared set of rules and limits. Some rules come from regulators (like Solvency II or the AI Act), others from the market or the firm's own governance. The model shows how each part makes decisions, while still aligning with the company's overall direction. This formalisation sets the stage for implementing a multi-agent system that reflects how real insurers actually function under legal, financial, and strategic constraints.

3 Functional Decomposition and Agent Architecture

Given the regulatory, confidentiality, and institutional constraints of the insurance sector, we argue that enterprise AI systems must be implemented as proprietary infrastructures rather than built on open, general-purpose platforms. This requirement arises from the intersection of data protection obligations (GDPR), auditability under regulatory supervision (IDD, AI Act), and the need for institution-specific objective functions. Within this context, we propose a decentralised multi-agent architecture as the appropriate system design, in which autonomous decision agents operate under a global regulatory and strategic logic enforced by an overseeing orchestrator agent.

The complexity of modern insurance firms necessitates a modular architecture, where distinct functions operate as autonomous, goal-directed subunits. This decomposition enhances both analytical tractability and system design, particularly when formalising insurer behaviour as a multi-agent system. Each agent is assigned a specific functional role, operating under local decision rules bounded by firm-level constraints. While agents such as underwriting, capital management, and claims settlement act semi-independently, they are coordinated through a global optimisation logic that enforces regulatory admissibility, risk exposure boundaries, and capital adequacy. Human-in-the-loop agents are integrated through a tiered access system, structuring data visibility and decision authority based on user roles, ensuring both regulatory compliance and operational alignment.

In addition to technical operations, the firm must satisfy ESG-related obligations, particularly in the social and governance (S&G) domains. This includes traceable decision accountability, inclusive stakeholder access, and transparent reporting structures. To this end, some agents may not optimise financial flows directly, but rather enforce compliance, track governance metrics, or interface with external actors. Large language models (LLMs) provide a natural interface layer for employee- and client-facing communication, enabling agents to interact with users in transparent, interpretable terms. These LLM-enabled agents can also serve stakeholders such as shareholders or regulators by offering on-demand conversational access to policy logic, performance summaries, and risk assessments. In this setting, natural language becomes both a user interface and a compliance instrument, supporting the broader governance objectives of the firm.

As a profit-oriented institution, the insurance firm seeks to maximise financial surplus subject to internal and regulatory constraints.

Capital Management Agent

The capital management agent is responsible for maintaining solvency while supporting business growth. Its function is to allocate capital across risk-bearing units, ensure compliance with solvency requirements (e.g. Solvency II), and buffer against adverse shocks. The agent does not generate profit directly but enables all other agents to operate within admissible financial boundaries. Its decision problem balances reserve levels, liquidity availability, and capital costs. Formally, the capital agent solves:

$$\max_{k \in \mathcal{K}} \quad -\gamma \cdot \text{Cost}(k) \tag{17}$$

$$\text{s.t.} \quad \mathbb{P}(X > k) \leq 1 - \alpha \quad (\text{Solvency constraint}) \tag{18}$$

$$k \geq \text{Reserve}_{\min} \quad (\text{Regulatory reserve floor}) \tag{19}$$

$$k \leq \text{AvailableCapital} \quad (\text{Capital availability}) \tag{20}$$

Here, k denotes the capital buffer allocated by the agent, and $\text{Cost}(k)$ is a convex function representing the opportunity cost or regulatory friction of holding capital. γ is a weighting coefficient that captures the trade-off between safety and efficiency. X is the random variable denoting aggregate liability. The solvency constraint (18) ensures that the probability of insolvency remains below a predefined risk threshold α (e.g. 99.5% for Solvency II). Constraint (19) imposes a reserve floor, and (20) limits decisions to available financial resources.

In plain terms, this agent decides how much capital the insurer should hold in reserve so that it can stay solvent during bad years, while also not tying up too much money that could otherwise be used for business. It plays a safety role in the system, making sure the company doesn't fall below legal or supervisory thresholds, while maintaining sufficient financial flexibility to support underwriting and operational activities.

Underwriting Agent

The underwriting agent is tasked with selecting, pricing, and classifying risks submitted to the insurer. Its goal is to accept profitable risks, reject adverse ones, and allocate fair premiums in line with expected losses

and capital requirements. This agent faces information asymmetries due to incomplete or biased disclosures by applicants, and must therefore rely on observable indicators or probabilistic models. It interacts closely with the pricing function, fraud detection, and capital allocation. Formally, the underwriting agent solves:

$$\max_{\mathbf{u} \in \mathcal{U}} \sum_{j=1}^m (p_j - \mathbb{E}[l_j \mid \mathbf{x}_j]) \cdot \delta_j \quad (21)$$

$$\text{s.t. } \delta_j \in \{0, 1\} \quad \forall j \quad (\text{Acceptance decision}) \quad (22)$$

$$\mathbb{E}[l_j \mid \mathbf{x}_j] \leq \tau \quad (\text{Expected loss threshold}) \quad (23)$$

$$\sum_{j=1}^m \delta_j \cdot \text{VaR}_\alpha(l_j) \leq k \quad (\text{Capital exposure constraint}) \quad (24)$$

In this formulation, j indexes the incoming applications, p_j is the proposed premium, l_j is the stochastic loss for applicant j , and $\mathbf{x}_j$ is the observable feature vector. δ_j is a binary variable that indicates whether the policy is accepted. The agent maximises expected underwriting profit across accepted applications while ensuring that no individual risk exceeds an expected loss threshold τ and that aggregate accepted risks do not violate the firm's capital limit k under VaR-based solvency requirements.

Put simply, this agent decides which applications to accept and at what premium. It tries to find a balance: avoiding customers likely to generate large losses, pricing correctly for those accepted, and keeping the portfolio within the firm's risk capacity. It acts as the firm's first line of financial defence.

Claims Handling Agent

The claims handling agent is responsible for assessing, verifying, and settling claims made by policyholders. Its role is to ensure fair and timely payment of legitimate claims while preventing overcompensation, delay-induced escalation, or exposure to fraudulent activity. It operates under uncertainty due to incomplete documentation and the stochastic nature of claim sizes and timing. It coordinates with the fraud detection agent, legal interface, and capital management unit. Formally, the claims agent solves:

$$\min_{\mathbf{c} \in \mathcal{C}} \sum_{i=1}^n \mathbb{E}[P_i(\theta_i)] + \lambda \cdot \text{Delay}_i(\theta_i) \quad (25)$$

$$\text{s.t. } P_i(\theta_i) \geq \underline{C}_i \quad \forall i \quad (\text{Minimum contractual payout}) \quad (26)$$

$$P_i(\theta_i) \leq \overline{C}_i \quad \forall i \quad (\text{Policy coverage limit}) \quad (27)$$

$$\text{FraudScore}_i(\theta_i) \leq \phi \quad \forall i \quad (\text{Fraud detection screen}) \quad (28)$$

Here, $P_i(\theta_i)$ is the payout for claim i as a function of latent claim type θ_i , which is uncertain at the time of processing. $\underline{C}_i$ and $\overline{C}_i$ denote the policy's minimum and maximum payout obligations. $\text{Delay}_i(\theta_i)$ represents cost associated with processing time (which may include penalties or legal escalation). The fraud score is an externally or internally computed index bounded by ϕ , above which claims are flagged for additional screening.

In simple terms, this agent manages the decision of how much to pay out and when. It tries to fulfil the firm's promises to policyholders but must be careful not to pay more than contractually required—or too quickly if the claim appears suspicious. Its job is about fairness, vigilance, and legal correctness under time pressure.

Fraud Detection Agent

The fraud detection agent operates in parallel with the claims and underwriting processes, monitoring for anomalous patterns or inconsistent declarations that may indicate intentional misrepresentation. Its purpose is to identify and flag high-risk cases for further review, thereby reducing exposure to internal and external manipulation. It leverages probabilistic models, anomaly detection algorithms, or learned patterns to assess the integrity of incoming data. This agent is preventive in nature and does not directly affect financial outcomes

but modifies the decision space of other agents. Formally, the fraud agent solves:

$$\max_{\mathbf{f} \in \mathcal{F}} \sum_{i=1}^n \mathbb{E}[\text{TP}_i(\theta_i)] - \mu \cdot \mathbb{E}[\text{FP}_i(\theta_i)] \quad (29)$$

$$\text{s.t. } \text{Score}_i(\theta_i) = \mathcal{M}(\mathbf{x}_i) \quad \forall i \quad (\text{Fraud model output}) \quad (30)$$

$$\text{Flag}_i = \mathbb{I}[\text{Score}_i(\theta_i) > \phi] \quad \forall i \quad (\text{Threshold decision}) \quad (31)$$

In this formulation, TP_i and FP_i represent the expected true and false positives for claim or policy i under the agent’s detection regime. μ is a penalty parameter on false positives to discourage excessive filtering. $\mathcal{M}(\cdot)$ is a predictive model mapping observable features $\mathbf{x}_i$ to a fraud score. Claims or applications are flagged if their score exceeds a risk threshold ϕ .

In simple terms, this agent watches for red flags. It uses data to guess which claims or applications might be fake or manipulated. If the score is too high, the case gets flagged for deeper review. It helps protect the insurer from bad actors without blocking genuine customers.

Compliance and Legal Agent

The compliance and legal agent ensures that all internal decisions and external product features adhere to applicable regulations and governance norms. It enforces admissibility under frameworks such as Solvency II, the Insurance Distribution Directive (IDD), the General Data Protection Regulation (GDPR), the AI Act, and ESG-related disclosure rules. This agent does not optimise profit directly, but constrains the action space of all other agents by validating whether decisions are legally and ethically permissible. Formally, the agent implements a filtering operator:

$$\forall i : \quad \mathbf{a}_i^{\text{final}} = \begin{cases} \mathbf{a}_i & \text{if } \mathbf{a}_i \in \mathcal{D}_{\text{legal}} \cap \mathcal{E}_{\text{ESG,E}} \cap \mathcal{G}_{\text{ESG,S\&G}} \\ \emptyset & \text{otherwise} \end{cases} \quad (32)$$

Here, $\mathbf{a}_i$ denotes the proposed action from agent i , and $\mathbf{a}_i^{\text{final}}$ is the admissible action after legal and ethical screening. The operator enforces exclusion if any rule—legal, environmental, social, or governance—is violated. ESG-S&G includes procedural fairness, transparency, stakeholder inclusion, and traceability of decisions.

In simple terms, this agent acts as the internal regulator. It checks whether actions suggested by other agents—such as new products, capital decisions, or underwriting rules—are legally allowed and ethically sound. If not, the action is blocked. It ensures the company plays by the rules and maintains reputational integrity.

Employee Interface Agent

The employee interface agent serves as a controlled gateway between personnel and the AI-based decision architecture of the insurance system. Its objective is to maximise the individual work utility of each employee by providing context-relevant access to agent outputs—such as those from underwriting, claims, or compliance—while enforcing strict internal control over data visibility, traceability, and policy adherence. Each employee $e \in \mathcal{E}$ operates within an organisational role $r \in \mathcal{R}$, which defines their permitted query set and data access tier. The agent facilitates system interaction through a filtered interface, while ensuring compliance with data protection obligations (e.g. GDPR), internal governance rules, and audit requirements. Formally, the agent solves the following constrained optimisation problem for each employee:

$$\max_{\mathbf{q}_e \in \mathcal{Q}_r} \mathbb{E}[U_e(\mathcal{L}(\mathbf{q}_e, \mathbf{a}_{\text{context}}^r))] \quad (33)$$

$$\text{s.t. } r \in \mathcal{R}_{\text{permitted}}(e) \quad (\text{Role-based access}) \quad (34)$$

$$\mathbf{a}_{\text{context}}^r \subseteq \mathbf{A}_{\text{global}} \quad (\text{Tier-constrained context}) \quad (35)$$

$$\log(\mathbf{q}_e, t, e) \in \mathcal{L}_{\text{audit}} \quad (\text{Supervisory auditability}) \quad (36)$$

Here, $\mathbf{q}_e$ denotes a query submitted by employee e , and $\mathbf{a}_{\text{context}}^r$ is the filtered view of the agent system state accessible under role r . The global state $\mathbf{A}_{\text{global}}$ comprises all structured outputs from other agents. The function $\mathcal{L}(\cdot)$ is the LLM interface that maps decision-relevant data to human-readable outputs. The utility function $U_e(\cdot)$ reflects the employee’s expected task efficiency or decision quality from receiving this response. Constraint (34) ensures that the role is authorised; (35) enforces tier-based data isolation; and (36) requires that every query is logged, timestamped, and auditable.

In simpler terms, this setup ensures that each employee gets exactly the information they need to do their job—no more, no less. Their role in the company defines what they are allowed to see. All their requests are checked, filtered, and recorded so that sensitive data stays protected and everything remains transparent and traceable. To operationalise the tiered system, the agent applies a conditional access mapping:

$$\text{Response}_e = \begin{cases} \mathcal{L}(\mathbf{q}_e, \mathbf{a}_{\text{context}}^r) & \text{if } r \in \mathcal{R}_{\text{permitted}}(e) \\ \emptyset & \text{otherwise} \end{cases} \quad (37)$$

This structure ensures that only admissible query–role combinations produce responses. For example, a call centre employee might access only claim-level summaries or policy status updates, whereas compliance officers can view audit logs, and executives are granted high-level system metrics and inter-agent coordination summaries. Role hierarchies are enforced dynamically through the internal access control engine, which references $\mathcal{R}_{\text{permitted}}(e)$ and monitors all activity under $\mathcal{L}_{\text{audit}}$. In functional terms, the employee interface agent enables precise, lawful, and productive human-AI interaction across the organisational hierarchy.

Put simply, the system checks who is asking and only gives an answer if the person’s role allows it. A junior employee might see only the basics needed for their task, while senior staff get broader insights. Everything is filtered automatically so that access always matches the person’s position and responsibility in the company.

Stakeholder Interface Agent

The stakeholder interface agent manages external access to the AI system by institutional and individual actors such as policyholders, regulators, auditors, shareholders, and business partners. Each stakeholder class is assigned a predefined visibility tier based on its contractual position, regulatory entitlement, or information rights. The agent ensures that external queries are resolved within lawful, contract-compliant, and context-specific boundaries, reflecting obligations under ESG-S&G standards—particularly explainability, procedural fairness, and inclusive governance. The objective of the stakeholder interface agent is to maximise the relevance and interpretability of information provided to authorised external users, without breaching confidentiality or exceeding regulatory limits. Each stakeholder $s \in \mathcal{S}$ is mapped to a visibility tier, which governs their admissible query set and contextual scope. Formally, the agent solves:

$$\max_{\mathbf{q}_s \in \mathcal{Q}_s} \mathbb{E}[U_s(\mathcal{L}(\mathbf{q}_s, \mathbf{a}_{\text{context}}^s))] \quad (38)$$

$$\text{s.t. } s \in \mathcal{S}_{\text{authorised}} \quad (\text{Stakeholder authentication}) \quad (39)$$

$$\mathbf{a}_{\text{context}}^s \subseteq \mathbf{A}_{\text{global}} \quad (\text{Tiered data scope}) \quad (40)$$

$$\log(\mathbf{q}_s, t, s) \in \mathcal{L}_{\text{audit}} \quad (\text{Audit trace}) \quad (41)$$

Here, $\mathbf{q}_s$ is a structured query submitted by stakeholder s , and $\mathcal{Q}_s$ is the set of queries permitted at that stakeholder’s access level. $\mathbf{a}_{\text{context}}^s$ is the filtered agent context available to that tier, constrained by $\mathbf{A}_{\text{global}}$, the complete internal state. The function $\mathcal{L}(\cdot)$ maps internal decisions into legally intelligible responses, adapted to jurisdiction, stakeholder contract, and purpose. The expected utility function $U_s(\cdot)$ captures the relevance and decision-usefulness of the returned information. Constraint (39) enforces access authentication; (40) restricts data exposure; and (41) ensures full supervisory traceability.

In simpler terms, this agent answers stakeholder questions using only the data they are legally allowed to see. It ensures that every external user—whether a customer, regulator, or investor—receives information that is relevant to their role, without exposing internal logic or private data. All interactions are tracked and audited to ensure lawful and accountable communication.

Stakeholder queries are processed under a conditional access mapping, where each stakeholder is assigned to a predefined access tier that determines which parts of the system they are permitted to view. This tiered structure reflects legal rights, contractual roles, and regulatory status, ensuring that each query is resolved only within the boundaries of the stakeholder’s assigned level.

$$\text{Response}_s = \begin{cases} \mathcal{L}(\mathbf{q}_s, \mathbf{a}_{\text{context}}^s) & \text{if } s \in \mathcal{S}_{\text{authorised}} \\ \emptyset & \text{otherwise} \end{cases} \quad (42)$$

This logic guarantees that each authorised stakeholder receives precisely the subset of information necessary for their role. A policyholder can access personal contracts, coverage parameters, and claim status. A regulator is entitled to solvency compliance data, ESG indicators, and audit trails. A shareholder may view profitability

breakdowns, capital allocation, and long-run portfolio structure. Smart contract enforcement and role registries dynamically govern access rights under $\mathcal{S}_{\text{authorised}}$.

Put simply, this rule acts like a smart filter. Only stakeholders with proper authorisation receive tailored answers to their questions. Everyone else gets nothing. This prevents unauthorised access and ensures that each user sees only what they are meant to—no more, no less.

Client Service Agent

The client service agent governs interactive access for existing policyholders, enabling personalised communication regarding policy coverage, claim progression, premium adjustments, renewals, and contractual terms. Its objective is to maximise customer clarity and decision-readiness while ensuring that all disclosures remain compliant with legal, contractual, and data protection boundaries. Each client $c \in \mathcal{C}$ is associated with a current contract, a policy data record $\mathbf{p}_c$, and a visibility tier based on policy status and jurisdiction. Formally, the client service agent solves:

$$\max_{\mathbf{q}_c \in \mathcal{Q}_c} \mathbb{E}[U_c(\mathcal{L}(\mathbf{q}_c, \mathbf{p}_c, \mathbf{a}_{\text{status}}))] \quad (43)$$

$$\text{s.t. } c \in \mathcal{C}_{\text{active}} \quad (\text{Contract status}) \quad (44)$$

$$\mathbf{p}_c \in \mathcal{P}_{\text{entitled}}(c) \quad (\text{Policy-bound visibility}) \quad (45)$$

$$\log(\mathbf{q}_c, t, c) \in \mathcal{L}_{\text{audit}} \quad (\text{Interaction traceability}) \quad (46)$$

Here, $\mathbf{q}_c$ is a query submitted by client c , and $\mathcal{Q}_c$ is the set of permitted queries under their active contract. The variable $\mathbf{p}_c$ represents their personalised policy attributes (e.g. coverage limits, deductibles, renewal terms), while $\mathbf{a}_{\text{status}}$ includes dynamic operational information such as claim updates or payment schedules. The LLM interface $\mathcal{L}(\cdot)$ converts structured internal data into accessible, legally compliant responses. The utility function $U_c(\cdot)$ reflects perceived information value, clarity, and support for action by the client. Constraints (44) and (45) limit visibility to entitled content only, while (46) ensures that all exchanges are auditable.

In simpler terms, the system checks whether a person is a valid, active customer and then tailors the response based on their contract. It helps them understand their rights, monitor their claims, and make informed choices—while keeping a clear record of every request and answer. To implement this logic, the agent applies a conditional access mapping:

$$\text{Response}_c = \begin{cases} \mathcal{L}(\mathbf{q}_c, \mathbf{p}_c, \mathbf{a}_{\text{status}}) & \text{if } c \in \mathcal{C}_{\text{active}} \\ \emptyset & \text{otherwise} \end{cases} \quad (47)$$

This structure ensures that only currently entitled policyholders receive access to personalised, contract-linked information. The visibility scope depends on the client’s policy type, current status (e.g. in-claim, pending renewal), and applicable legal protections. For example, a health insurance client in an open claim may query reimbursement timelines, while a property policyholder near renewal can request premium comparisons. The filtering logic reflects a tiered access model tied to dynamic contract context and compliance policy.

Put plainly, this agent acts like a personalised digital advisor: it checks if someone is still an active client and then answers questions based only on what they are allowed to see. It keeps things clear, relevant, and legal—without ever showing what doesn’t belong to them.

Client Acquisition Agent

The client acquisition agent is responsible for engaging potential customers, mapping product offerings to prospect profiles, and pre-filtering applications for underwriting relevance. It operates upstream of the underwriting agent and integrates with marketing, pricing, and legal modules. Its role is to generate qualified leads, screen suitability, and deliver regulatory pre-contractual information in accessible language. The agent is constrained by marketing compliance rules and fairness requirements under anti-discrimination directives.

Formally, the agent implements a pre-qualification mapping:

$$\mathbf{p}_{\text{suggested}} = \mathcal{M}_{\text{map}}(\mathbf{x}_{\text{prospect}}) \quad \text{if } \mathbf{x}_{\text{prospect}} \in \mathcal{X}_{\text{admissible}} \quad (48)$$

Here, $\mathbf{x}_{\text{prospect}}$ contains declared or inferred features of a potential customer (e.g. age, location, coverage needs), and $\mathcal{M}_{\text{map}}(\cdot)$ maps these features to a product recommendation $\mathbf{p}_{\text{suggested}}$, subject to legal filters

$\mathcal{X}_{\text{admissible}}$ (e.g. no discriminatory profiling, marketing restrictions). The agent also exposes LLM interfaces for interactive product comparison, consent dialogue, and onboarding FAQs.

In effect, this agent serves as the public face of the firm’s system, converting prospects into compliant applications. It educates, filters, and prepares the ground for underwriting by delivering structured pre-application guidance, all while ensuring that acquisition processes are transparent, equitable, and regulatorily compliant.

System Orchestrator Agent

The system orchestrator agent serves as the supervisory layer responsible for ensuring global consistency, strategic alignment, and institutional coherence across all subordinate agents. It oversees multi-agent coordination, verifying that local decisions—while independently optimised—collectively satisfy the legal, financial, and operational constraints of the organisation. The orchestrator does not perform direct optimisation; instead, it acts as a global validator of admissibility, equilibrium, and inter-agent alignment. Formally, the orchestrator monitors whether the system-wide agent state $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ respects both the global objective structure and admissibility envelope:

$$\{\mathbf{a}_i\}_{i=1}^n \in \arg \max_{\mathbf{a}_1, \dots, \mathbf{a}_n} \sum_i \mathbb{E}[U_i(\pi_i(\mathbf{a}_i, \mathbf{a}_{-i}))] \quad \text{s.t. global feasibility and coordination constraints} \quad (49)$$

Its role is to verify system-wide admissibility and consistency and to authorise or reject execution based on validation:

$$\text{Approved} = \begin{cases} 1 & \text{if all admissibility, consistency, and coordination rules are met} \\ 0 & \text{otherwise} \end{cases} \quad (50)$$

The orchestrator integrates several supervisory mechanisms to enforce institutional coherence. First, it performs constraint propagation across agent boundaries, ensuring that local admissibility conditions do not conflict when aggregated at the system level. Second, it verifies role-permission consistency across agents, maintaining uniformity in how organisational roles are mapped to actions and data access within different subsystems. Third, it monitors for cross-agent conflicts—such as contradictory objectives, redundant actions, or mutual constraint violations—and triggers resolution logic when necessary. Finally, the orchestrator aligns internal agent decisions with external supervisory expectations and the firm’s strategic objectives, ensuring that the AI system acts in accordance with institutional goals and regulatory mandates.

It may override, veto, or delay decisions that would introduce institutional incoherence, constraint violations, or regulatory breaches. While the legal/compliance agent enforces rule-level admissibility for individual actions, the orchestrator governs **system-level integration** and strategic consistency. LLM modules attached to this agent may generate system summaries, policy explanations, or audit statements for institutional stakeholders (e.g. board, regulators).

In simpler terms, this agent makes sure that the AI system behaves like one coordinated organisation—not a set of disconnected bots. It checks that all decisions fit together, follow the rules, and support the company’s overall strategy before anything gets executed.

4 Unified Agent System and Communication Logic

To complete the agent-based system architecture, we introduce a unified optimisation structure that integrates both internal decision agents and human interface agents within a single analytical expression. This formulation provides a consistent basis for verifying system-wide admissibility, equilibrium, and institutional alignment. It enables the orchestration layer to assess the joint admissibility of all agent actions and user queries before any execution occurs. The integration of human-in-the-loop components into the agent architecture does not require the introduction of a separate communication agent. Instead, the dedicated interface agents—employee, stakeholder, and client—already serve as structured access points under role-based governance. These agents enforce strict admissibility rules through data filtering, role-tier mappings, and audit logs. Their outputs are routed through the System Orchestrator Agent, which validates whether all resulting actions and queries cohere with the institutional constraints of the firm. This routing logic preserves modular clarity while avoiding the control fragmentation that would result from parallel coordination layers. Introducing an additional communication core would dilute accountability, obscure institutional logic, and conflict with the

compliance-by-design principles underlying the system. Formally, the global optimisation problem can be written as:

$$\max_{\{\mathbf{a}_i\}, \{\mathbf{q}_j\}} \sum_{i=1}^n \mathbb{E}[U_i(\pi_i(\mathbf{a}_i, \mathbf{a}_{-i}))] + \sum_{j=1}^m \mathbb{E}[V_j(\mathcal{L}(\mathbf{q}_j, \boldsymbol{\alpha}_j))] \quad (51)$$

$$\text{s.t. } \mathbf{a}_i \in \mathcal{D}_i \cap \mathcal{E}_i \cap \mathcal{G}_i \cap \mathcal{O}_i, \quad \forall i \quad (52)$$

$$\mathbf{q}_j \in \mathcal{Q}_j, \quad \boldsymbol{\alpha}_j \subseteq \mathbf{A}_{\text{global}}, \quad r_j \in \mathcal{R}_{\text{permitted}}(j), \quad \forall j \quad (53)$$

$$\{\mathbf{a}_i\}, \{\mathbf{q}_j\} \in \mathcal{A}_{\text{valid}} \quad (\text{system-level admissibility verified by orchestrator}) \quad (54)$$

Here, $\mathbf{a}_i$ denotes the action vector of internal agent i and $\pi_i(\cdot)$ is the agent-specific decision function, potentially dependent on the actions $\mathbf{a}_{-i}$ of other agents. The function $U_i(\cdot)$ measures the expected utility of the action, incorporating performance, compliance, or risk-based criteria. Each query $\mathbf{q}_j$ is submitted by a human-facing interface agent j and evaluated through a large language model $\mathcal{L}(\cdot)$ applied to a filtered context $\boldsymbol{\alpha}_j$, drawn from the global agent state $\mathbf{A}_{\text{global}}$. The function $V_j(\cdot)$ captures the informational or operational utility returned to the user. The admissibility conditions restrict each internal action to the intersection of legally, environmentally, socially, and operationally permitted domains. Interface queries must be issued under valid role authorisations r_j , conform to data access constraints, and remain fully traceable. The orchestrator confirms that the combined set of actions and queries is admissible, non-contradictory, and institutionally consistent.

In simple terms, this equation brings all agent decisions—both machine-made and human-triggered—into one system-wide logic. Each agent tries to do its job well, and each user-facing agent responds only within allowed limits. Nothing proceeds until the orchestrator has checked that every action and answer fits together without breaking any legal or operational rule. This avoids having too many control layers and ensures the AI system behaves like one coherent institution.

5 Protocols and Agent Integration in Enterprise AI

The deployment of multiple agents in enterprise AI arises from the functional decomposition of insurance firms into distinct domains—capital allocation, underwriting, compliance, and claims handling—each governed by specific constraints, data privileges, and decision logic. Decentralisation alone is insufficient. To ensure coherence, regulatory admissibility, and shared utility optimisation, agents must be coordinated via protocol layers that standardise context propagation, state transfer, and inter-agent reasoning. This necessitates interoperability frameworks such as the Model Context Protocol (MCP) and Agent-to-Agent (A2A) messaging standards. Enterprise AI systems operate asynchronously across heterogeneous components with divergent compute schedules, partial observability, and uneven access to real-time data. Synchronous coordination introduces latency and fragility. Asynchronous architectures mitigate these issues by decoupling data availability from execution. Tadi [21] shows that asynchronous data handling enhances Progressive Web Application robustness, supporting offline operation and non-blocking updates—critical for agent continuity in intermittently connected environments. At the training level, asynchronous federated learning enables model updates without central synchronisation [22], reinforcing system resilience. Beyond architectural theory, empirical studies demonstrate that agent communication latency increases with message size and agent count. Berna-Koes et al. [23] show that message delays can grow non-linearly in multi-agent networks unless efficient backchannel protocols are implemented. Further, in delay-sensitive coordination tasks, latency-aware communication models such as DACOM [24] significantly enhance performance by adapting decision logic to inter-agent delays. Lei et al. [25] provide a framework for synchronising asynchronous perceptual features in collaborative systems, improving agent robustness in high-latency environments. In regulated domains such as insurance, empirical auditability standards further constrain system design. The UK Financial Conduct Authority requires AI systems to provide traceable decision logs within 24–72 hours of execution [26]. The Financial Stability Board emphasises the importance of explainability and audit resilience to manage systemic risks in AI-driven financial services [27]. These empirical thresholds reinforce the need for persistent state management and protocol-level memory that can support post-hoc review under institutional supervision. The Agent-to-Agent (A2A) protocol, introduced by Google, standardises inter-agent messaging across frameworks such as LangGraph, CrewAI, and AutoGen by defining shared syntax for intention, memory, and task state. A2A allows agents built in distinct runtime environments to exchange structured content—decisions, delegation requests, alerts—while decoupling role semantics from implementation. In heterogeneous enterprise settings, where agents vary in logic paradigms

or regulatory models, A2A enables coordination between, for example, an LLM-based compliance module and a rule-based pricing agent. As noted in [28], A2A promotes composability by providing a shared abstraction for communication, aligning objectives under constrained interoperability. While A2A addresses syntactic and messaging compatibility, the Model Context Protocol (MCP) governs semantic coherence and memory persistence. MCP structures context propagation and state retention, enabling agents to reason over shared histories, hierarchical goals, and environmental signals. Krishnan [29] highlights MCP’s capacity to encode long-range dependencies, allowing agents to retain mission-critical knowledge—e.g., regulatory thresholds, prior decisions, or user interactions—vital in financial or legal workflows. MCP functions as the institutional memory layer, necessary for consistency, justification of contingent decisions, and compliance transparency. Narajala and Habler [30] further underscore MCP’s role in auditability, enabling ESG alignment and AI Act compliance by preserving reasoning chains and allowing decision context replay. A2A and MCP are not substitutes but complements: A2A ensures technical interoperability and communication reliability; MCP secures semantic alignment and persistent memory. Together, they constitute a dual-stack foundation for robust, auditable multi-agent systems. Protocol-level design is thus not a technical convenience but a regulatory and operational necessity. The Agent-to-Agent (A2A) protocol, introduced by Google, standardises inter-agent messaging across frameworks such as LangGraph, CrewAI, and AutoGen by defining shared syntax for intention, memory, and task state [31]. A2A allows agents built in distinct runtime environments to exchange structured content—decisions, delegation requests, alerts—while decoupling role semantics from implementation. In heterogeneous enterprise settings, where agents vary in logic paradigms or regulatory models, A2A enables coordination between e.g., an LLM-based compliance module and a rule-based pricing agent. As noted in [28], A2A promotes composability by providing a shared abstraction for communication, aligning objectives under constrained interoperability. While A2A addresses syntactic and messaging compatibility, the Model Context Protocol (MCP) governs semantic coherence and memory persistence. MCP structures context propagation and state retention, enabling agents to reason over shared histories, hierarchical goals, and environmental signals [32]. Krishnan [29] highlights MCP’s capacity to encode long-range dependencies, allowing agents to retain mission-critical knowledge—e.g., regulatory thresholds, prior decisions, or user interactions—vital in financial or legal workflows. MCP functions as the institutional memory layer, necessary for consistency, justification of contingent decisions, and compliance transparency. Narajala [30] further underscores MCP’s role in auditability, enabling ESG alignment and AI Act compliance by preserving reasoning chains and allowing decision context replay. A2A and MCP are not substitutes but complements: A2A ensures technical interoperability and communication reliability; MCP secures semantic alignment and persistent memory. Together, they constitute a dual-stack foundation for robust, auditable multi-agent systems [32, 31]. MCP ensures that agent interactions are contextually meaningful; A2A ensures they remain operable across environments.

Protocol-level design is thus central to enterprise AI. The insurance architecture proposed here builds upon this dual-protocol paradigm, combining asynchronous operation with explainable, persistent decision memory to meet institutional and regulatory demands.

6 Conclusion

This paper presents a formal architecture for implementing enterprise AI in regulated insurance firms, grounded in firm theory, constrained optimisation, and institutional logic. The insurer is modelled as a structured decision system subject to solvency constraints, legal admissibility, ESG obligations, and internal feasibility limits. This framework enables a principled decomposition of the firm’s behaviour into analytically distinct subfunctions aligned with regulatory and strategic demands.

A multi-agent architecture defines each functional domain—capital management, underwriting, claims, and compliance—as a bounded rational agent operating within a shared constraint environment. Local decision models integrate directly into the firm’s institutional structure, ensuring that optimisation occurs strictly within admissible legal, financial, and ESG boundaries.

An orchestrator agent enforces system-level coherence, validating inter-agent alignment, propagating global constraints, and maintaining institutional consistency. Protocol-level infrastructures such as asynchronous coordination, memory persistence, and semantic interoperability via MCP and A2A support structured communication across heterogeneous agent modules.

This framework advances the structural modelling of regulated firms by embedding regulatory, operational, and governance constraints into the decision logic itself. Unlike conventional models that treat compliance as an external layer, this architecture integrates institutional constraints into core decision processes, forming a foundation for building auditable, compliant AI systems in high-stakes institutional domains where formal

accountability and systemic coherence are essential.

References

1. DiMaggio PJ and Powell WW. The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields. *American Sociological Review* 1983; 48:147–60. DOI: [10.2307/2095101](https://doi.org/10.2307/2095101)
2. Boonen TJ. Solvency II solvency capital requirement for life insurance companies based on expected shortfall. *European Actuarial Journal* 2017; 7:405–34. DOI: [10.1007/s13385-017-0160-4](https://doi.org/10.1007/s13385-017-0160-4)
3. Schmeiser H and Gründl H. Pricing Double-Trigger Reinsurance Contracts. Financial Versus Actuarial Approach. *SSRN Electronic Journal* 2002. DOI: [10.2139/ssrn.302697](https://doi.org/10.2139/ssrn.302697)
4. Biener C and Eling M. Organization and Efficiency in the International Insurance Industry: A Cross-Frontier Analysis. *European Journal of Operational Research* 2012; 221:454–68. DOI: [10.1016/j.ejor.2012.03.037](https://doi.org/10.1016/j.ejor.2012.03.037)
5. Arrow KJ. Uncertainty and the Welfare Economics of Medical Care. *American Economic Review* 1963; 53:941–73
6. Rothschild M and Stiglitz J. Equilibrium in Competitive Insurance Markets: An Essay on the Economics of Imperfect Information. *Quarterly Journal of Economics* 1976; 90:629–49. DOI: [10.2307/1885326](https://doi.org/10.2307/1885326)
7. Zweifel P and Eisen R. *Insurance Economics*. Springer Texts in Business and Economics. Springer, 2012. DOI: [10.1007/978-3-642-20548-4](https://doi.org/10.1007/978-3-642-20548-4)
8. Dickson DC and Drekić S. The Joint Distribution of the Surplus Prior to Ruin and the Deficit at Ruin in Some Sparre Andersen Models. *Insurance: Mathematics and Economics* 2004; 34:425–34. DOI: [10.1016/j.insmatheco.2003.11.003](https://doi.org/10.1016/j.insmatheco.2003.11.003)
9. Cummins JD and Weiss MA. Organizational Form and Efficiency: The Coexistence of Stock and Mutual Property-Liability Insurers. *Management Science* 1999; 45:1254–69. DOI: [10.1287/mnsc.45.9.1254](https://doi.org/10.1287/mnsc.45.9.1254)
10. Brockett PL, Xia X, and Derrig RA. Fraud Classification Using Principal Component Analysis of RIDITs. *Journal of Risk and Insurance* 2002; 69:325–50. DOI: [10.1111/1539-6975.00027](https://doi.org/10.1111/1539-6975.00027)
11. Kolambe S and Kaur P. Survey on Insurance Claim Analysis Using Natural Language Processing and Machine Learning. *International Journal on Recent and Innovation Trends in Computing and Communication* 2023; 11:1–7. Available from: <https://ijritcc.org/index.php/ijritcc/article/view/8461>
12. Automation Anywhere. *Intelligent Automation for Insurance Claims Processing*. Automation Anywhere 2021. Available from: <https://www.automationanywhere.com/solutions/insurance>
13. Starita MG and Malafronte I. *Capital Requirements, Disclosure, and Supervision in the European Insurance Industry*. Palgrave Macmillan, 2014. DOI: [10.1057/9781137390844](https://doi.org/10.1057/9781137390844)
14. Rae RA, Barrett A, Brooks D, Chotai MA, Pelkiewicz AJ, and Wang C. A review of Solvency II: Has it met its objectives? *British Actuarial Journal* 2017; 22:1–32. DOI: [10.1017/S1357321717000241](https://doi.org/10.1017/S1357321717000241)
15. European Insurance and Occupational Pensions Authority. *Report on the Application of the Insurance Distribution Directive (IDD)*. 2022. Available from: https://www.eiopa.europa.eu/system/files/2022-01/eiopa-bos-21-581_report_on_the_application_of_the_idd.pdf

16. Werner T. The IDD and Its Impact on the Life Insurance Industry. *The Future of Insurance Regulation in the European Union*. Ed. by Kumpan C and Fleckner AM. Springer, 2021 :85–102. DOI: [10.1007/978-3-030-52738-9_4](https://doi.org/10.1007/978-3-030-52738-9_4)
17. European Commission. Proposal for a Regulation Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). 2021. Available from: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206>
18. European Parliament and Council. Regulation (EU) 2019/2088 on Sustainability-related Disclosures in the Financial Services Sector (SFDR). 2020. Available from: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R2088>
19. Parliament E and Council. Regulation (EU) 2020/852 on the Establishment of a Framework to Facilitate Sustainable Investment (EU Taxonomy). 2020. Available from: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32020R0852>
20. Parliament E and Council. Regulation (EU) 2019/2088 on Sustainability-Related Disclosures in the Financial Services Sector (SFDR). 2019. Available from: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R2088>
21. Tadi SRCCT. Asynchronous Data Processing with Machine Learning Progressive Web Applications for Seamless Offline Functionality. *International Journal of Science and Research* 2022; 11:1288–96. DOI: [10.21275/SR220912120313](https://doi.org/10.21275/SR220912120313)
22. Hou X, Zhao Y, Wang S, and Wang H. Make Robust Asynchronous Federated Learning Possible. arXiv preprint arXiv:2504.04081 2024. DOI: [10.48550/arXiv.2504.04081](https://doi.org/10.48550/arXiv.2504.04081)
23. Berna-Koes D, Simmons R, and Veloso M. Communication Efficiency in Multi-Agent Systems. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2004 :1228–35. Available from: <https://www.cs.cmu.edu/~softagents/papers/paper778.pdf>
24. Yuan Y, Yang T, Cai Q, Wen P, Wang L, Yang W, and Ye D. DACOM: Learning Delay-Aware Communication for Multi-Agent Reinforcement Learning. arXiv preprint. 2022. arXiv: [2212.01619](https://arxiv.org/abs/2212.01619). Available from: <https://arxiv.org/abs/2212.01619>
25. Lei Y, Zhu X, Liu Z, Wang Z, Wang Y, Huang G, Sun M, and Lin D. Latency-Aware Collaborative Perception. arXiv preprint. 2022. arXiv: [2207.08560](https://arxiv.org/abs/2207.08560). Available from: <https://arxiv.org/abs/2207.08560>
26. Authority FC. Artificial Intelligence Update. Tech. rep. UK Financial Conduct Authority, 2022. Available from: <https://www.fca.org.uk/publication/corporate/ai-update.pdf>
27. Board FS. Artificial Intelligence and Machine Learning in Financial Services: Market Developments and Financial Stability Implications. Tech. rep. Financial Stability Board, 2017. Available from: <https://www.fsb.org/uploads/P011117.pdf>
28. Analytics Vidhya Editorial Team. Agent to Agent Protocol Helping AI Agents Work Together. <https://community.analyticsvidhya.com/c/generative-ai-tech-discussion/agent-to-agent-protocol-helping-ai-agents-work-together-across-systems>. 2025
29. Krishnan N. Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications. arXiv preprint. 2025. arXiv: [2504.21030](https://arxiv.org/abs/2504.21030). Available from: <https://arxiv.org/abs/2504.21030>
30. Narajala VS and Habler I. Enterprise-Grade Security for the Model Context Protocol (MCP): Frameworks and Mitigation Strategies. arXiv preprint. 2025. DOI: [10.48550/arXiv.2504.08623](https://doi.org/10.48550/arXiv.2504.08623). arXiv: [2504.08623](https://arxiv.org/abs/2504.08623). Available from: <https://arxiv.org/abs/2504.08623>
31. Google Developers. Announcing the Agent2Agent Protocol (A2A). <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>. 2025

32. Hou X, Zhao Y, Wang S, and Wang H. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. arXiv preprint arXiv:2503.23278 2024. DOI: [10.48550/arXiv.2503.23278](https://doi.org/10.48550/arXiv.2503.23278)