http://dx.doi.org/10.18623/rvd.v23.n4.4344

API ECOSYSTEMS IN THE AGE OF ARTIFICIAL INTELLIGENCE

ECOSSISTEMAS DE API NA ERA DA INTELIGÊNCIA ARTIFICIAL

Article received on: 10/3/2025
Article accepted on: 1/5/2026

Félix Témolé*
*Chief Information Officer (CIO) Advisory Services, Capgemini Germany GmbH, Hamburg, Germany
Orcid: https://orcid.org/0009-0007-7815-8800
felix.temole@capgemini.com

Desislava Atanasova**
**Department of Informatics and Information Technologies (IIT), University of Ruse “Angel, Kanchev”, Ruse, Bulgaria
Orcid: https://orcid.org/0000-0001-7147-3890
datanasova@uni-ruse.bg

The authors declare that there is no conflict of interest

Abstract

APIs are undergoing a fundamental shift from static integration mechanisms toward dynamic, AI-interpretable interaction surfaces. Large language models and autonomous agents increasingly discover, understand, and orchestrate APIs with minimal human intervention, reshaping integration paradigms across domains. This systematic review (2020–2026) analyzes emerging AI-native API ecosystems along six dimensions: functionality, security, governance, architecture, efficiency, and application areas. The findings highlight an evolution from conventional REST, SOAP, and messaging architectures to adaptive, context-aware, and policy-driven interface models. Concurrently, novel security risks—such as prompt injection, model manipulation, and cascading threats in multi-layer API orchestrations—intensify the need for advanced protective controls, including mTLS, OAuth 2.1, and zero-trust governance architectures. A key contribution of this work is a taxonomy that classifies AI-driven API ecosystems according to autonomy level, governance maturity, interoperability, security posture, and energy efficiency. The review positions APIs as foundational components of intelligent systems and offers guidance for research, standardization efforts, and the secure deployment of AI-native API architectures.

Keywords: API Ecosystems. Artificial Intelligence. Agentic AI. API Security. Governance. Interoperability. Energy Efficiency.

Resumo

As APIs estão passando por uma mudança fundamental, passando de mecanismos de integração estáticos para superfícies de interação dinâmicas e interpretáveis por IA. Grandes modelos de linguagem e agentes autônomos descobrem, compreendem e orquestram cada vez mais as APIs com intervenção humana mínima, remodelando os paradigmas de integração em todos os domínios. Esta revisão sistemática (2020-2026) analisa os ecossistemas emergentes de APIs nativas de IA em seis dimensões: funcionalidade, segurança, governança, arquitetura, eficiência e áreas de aplicação.

As conclusões destacam uma evolução das arquiteturas REST, SOAP e de mensagens convencionais para modelos de interface adaptáveis, sensíveis ao contexto e orientados por políticas. Simultaneamente, novos riscos de segurança — como injeção de prompt, manipulação de modelos e ameaças em cascata em orquestrações de API multicamadas — intensificam a necessidade de controles de proteção avançados, incluindo mTLS, OAuth 2.1 e arquiteturas de governança zero trust. Uma contribuição importante deste trabalho é uma taxonomia que classifica os ecossistemas de API orientados por IA de acordo com o nível de autonomia, maturidade de governança, interoperabilidade, postura de segurança e eficiência energética. A revisão posiciona as APIs como componentes fundamentais dos sistemas inteligentes e oferece orientação para pesquisa, esforços de padronização e implantação segura de arquiteturas de API nativas de IA.

Palavras-chave: *Ecossistemas de API. Inteligência Artificial. IA Agente. Segurança de API. Governança. Interoperabilidade. Eficiência Energética.*

1 INTRODUCTION

APIs form the backbone of digital platforms across domains such as finance, healthcare, public administration, and mobility, enabling interoperability, data exchange, and service orchestration at scale [1], [2]. With the rise of large language models (LLMs) and agent-based frameworks, a fundamental shift is underway: APIs are evolving from static integration contracts into dynamic, semantically interpretable interaction surfaces that AI systems can discover and utilize autonomously [3], [4], [5], [6]. LLM-driven agents increasingly plan, select, and orchestrate tool/API calls—moving from human-only development models toward hybrid or fully autonomous digital actors—which, in turn, elevates requirements for transparency, traceability, auditability, and regulatory compliance [3], [6].

Recent surveys of agentic AI highlight architectures, communication mechanisms, memory models, and safety boundaries, while noting persistent gaps in portability, reproducibility, evaluation rigor, and robust trust models [3], [4], [7]. In parallel, established interface and observability standards continue to mature. For HTTP APIs, studies of OpenAPI-based ecosystems and automated description generation show the consolidation of specification-driven design and tooling that supports adaptive, policy-aware interfaces [8], [9]. For service communication, empirical evaluations consistently find gRPC/HTTP-2—and increasingly HTTP/3/QUIC—delivering latency/throughput advantages under microservice loads, informing architectural choices for high-efficiency API stacks [10], [11], [12]. Meanwhile, OpenTelemetry is being adopted as a unified, vendor-neutral instrumentation standard; academic and conference work analyzes semantic alignment and deployment trade-offs, including configuration impacts on overhead and stability in distributed systems [13], [14], [15].

Regulatory frameworks are also advancing. Scholarly analyses of the EU AI Act (Regulation (EU) 2024/1689) examine its risk-based approach, governance instruments, and interplay with adjacent EU digital legislation [16], [17], [18]. In organizational AI governance, ISO/IEC 42001 is emerging as the first AI-specific management-system standard, with academic commentary and monographs mapping its controls to legal and ethical requirements [19], [20]. Complementing these, research proposes maturity models and operational guidance that align enterprise practice with the NIST AI Risk Management Framework (AI RMF 1.0) [21].

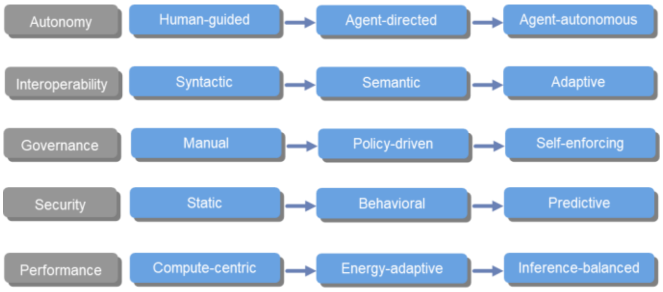
Security risks in AI-native API orchestration are growing. Work on prompt injection and multi-agent cascading attacks demonstrates how adversarial instructions can propagate across tools and agents, defeating naive guardrails and causing system-wide disruptions [22], [23], [24]. In response, the literature emphasizes defense-in-depth controls at the API and transport layers—e.g., mutual TLS (mTLS) for bidirectional service authentication, OAuth-based authorization with modern best practices (e.g., PKCE; deprecation of insecure flows), and Zero-Trust control architectures for continuous verification and least-privilege enforcement [25], [26], [27], [28].

Against this backdrop, this work makes five key contributions:

1. A structured and simplified taxonomy of AI-driven API ecosystems (*Figure 1*) that classifies systems by autonomy level, interoperability, governance maturity, security posture, and energy efficiency—grounded in recent agentic-AI surveys and tool-use benchmarks [3], [5], [6] .
2. A reference architecture unifying OpenAPI-modeled HTTP endpoints with gRPC/HTTP-3 communication and OpenTelemetry observability, emphasizing adaptive, context-aware, and policy-driven interfaces [8], [10], [13].
3. A governance mapping that systematically compares compliance obligations and control families across the EU AI Act, ISO/IEC 42001, and NIST AI RMF—highlighting alignment and practical adoption pathways [16], [19], [21].
4. A synthesis of current security insights on prompt injection, agent tool misuse, and technical mitigations spanning mTLS, OAuth-based authorization, and Zero-Trust reference models [22], [23], [24], [25], [26], [27], [28].
5. A PRISMA-2020-compliant methodology for a systematic review (2020–2026), including documented identification, screening, eligibility, and inclusion criteria [29], [30].

Figure 1

Taxonomy of AI-Driven API Ecosystems (simplified) (Source: Own Representation)



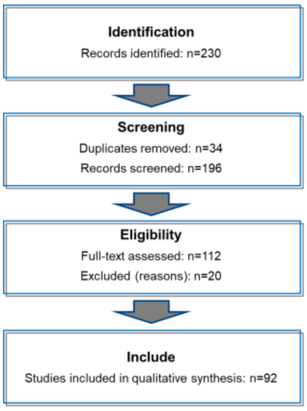
2 METHODOLOGY

This work is based on a Systematic Literature Review (SLR) conducted in accordance with the PRISMA 2020 guidelines, which define standardized procedures for identification, screening, eligibility assessment, and synthesis of scientific evidence [29], [30]. The search was executed across leading scholarly databases—including IEEE Xplore, ACM Digital Library, SpringerLink, and arXiv—and complemented with curated industry white-paper collections. The search window covered the years 2016 to 2026, with a specific analytical focus on 2020–2026, reflecting the rapid acceleration of LLM-driven agent systems, AI-enabled API ecosystems, and emerging security requirements [3], [4]. In total, 230 publications were identified; after title/abstract screening, full-text review, and quality appraisal, 92 sources were included in the final synthesis.

A simplified visualization of the PRISMA flow is shown below (*Figure 2*).

Figure 2

PRISMA Flow Diagram (simplified) (Source: Own Representation)



2.1 Search strategy

The search strategy consisted of multi-stage Boolean queries targeting AI-supported API ecosystems, guided by recent advances in API specifications (e.g., OpenAPI), high-performance service communication (e.g., gRPC), and unified observability standards (e.g., OpenTelemetry) [8], [9], [10], [11], [13], [14], [15].

The core search terms consisted of a combined Boolean expression targeting AI-enabled API ecosystems, formulated as (“API ecosystems” OR “OpenAPI” OR “gRPC” OR “OpenTelemetry”) AND (“LLM” OR “agent” OR “autonomous” OR “prompt injection”)

The search was limited to Publications in English, the period 2020–2026, scientifically documented or technically substantiated sources (peer-reviewed studies, methodologically transparent preprints, and academically grounded technical reports).

The documentation of search runs—including identification paths, database filters, and exclusion reasons—adhered strictly to PRISMA 2020 requirements [29], [30].

2.2 Inclusion and exclusion criteria

2.2.1 Inclusion criteria

- Research or technical reports with direct relevance to AI-supported, AI-interacting, or AI-dependent API ecosystems, consistent with growing agent-based and tool-augmented API research [3], [4], [10].
- Peer-reviewed studies, transparent preprints, and high-quality analytical reports.
- Works addressing API technologies, LLM–API interaction, autonomous agents, or associated governance/security mechanisms, including prompt-injection-related risks [22], [23] [24].

2.2.2 Exclusion criteria

- Duplicates or redundant publications,

- Purely theoretical works without an API or AI connection,
- Non-English publications,
- Sources lacking methodological transparency or traceability.

2.3 Data extraction and quality assessment

Data extraction was performed by two independent reviewers, capturing:

- Publication year and type,
- Thematic category and research domain,
- API technologies employed (e.g., OpenAPI, gRPC, telemetry systems),
- Underlying AI models or agent architectures,
- Evaluation methods, risks, and reported system properties.

Quality assessment followed a PRISMA-compliant evaluation scheme, emphasizing:

- Methodological transparency,
- Reproducibility,
- External validity and transferability,
- Clear articulation of assumptions and limitations.

This framework aligns with rigor standards commonly applied in AI-driven agent research, observability studies, and microservice/API evaluation literature [10], [13], [14], [15].

2.4 Synthesis method

Synthesis followed a qualitative thematic approach across the major categories:

- Functionality,
- Security,
- Governance,
- Architecture,
- Efficiency,
- Cross-domain applications.

Due to the inherent heterogeneity among included studies—stemming from diverse technologies, evaluation methodologies, and maturity levels—no meta-analysis was conducted. Instead, all sources were organized into qualitative clusters to reveal structural patterns and research trajectories, as recommended in contemporary reviews of LLM-based agents, API specification ecosystems, and prompt-injection security research [3], [8], [23], [22], [24].

Aggregated results from identification, title/abstract screening, and full-text assessment are summarized in (*Table 1*).

Table 1

Review Summary Table (Source: Own Representation)

Category	Description
Review Type	Systematic Literature Review (SLR) following PRISMA 2020 [29], [30]
Databases / Sources	IEEE Xplore, ACM Digital Library, SpringerLink, arXiv, curated white-paper repositories
Timeframe	Initial coverage 2016–2026; focused analytical window 2020–2026 [3], [4]
Identified Sources	230 publications
Included Studies	92 studies after screening and quality evaluation
Search Strtegy	Boolean queries targeting API ecosystems & LLM-/agent-based systems [3], [8], [15], [23], [22], [24]
Inclusion Criteria	Peer-reviewed or technically rigorous sources on AI-supported API ecosystems
ExclusionCriteria	Duplicates, non-relevant theoretical work, non-English publications, insufficient methodology
Data Extraction	Conducted by two independent reviewers; extracted metadata, technologies, AI models, evaluation methods
Quality Assessment	PRISMA-aligned scheme emphasizing transparency, reproducibility, external validity
Synthesis	Qualitative thematic synthesis; no meta-analysis due to heterogeneity

3 EVOLUTION OF API ECOSYSTEMS IN THE AGE OF AI

The development of modern API ecosystems can be divided into three major evolutionary phases, each shaped by technological paradigm shifts, changing interaction models, and new requirements for flexibility, security, and governance. Recent academic literature confirms that the evolution of APIs increasingly intersects with advances in large language models (LLMs), agentic AI, and semantically enriched interface design [31], [32], [33], [34].

3.1 Phase 1: classical API architectures

In the first evolutionary phase, established architectural patterns such as REST, SOAP, and message-oriented services dominated. These interfaces relied on static schemas, manually written documentation, and direct, largely manual interaction between developers and systems. Although these APIs were robust and stable, they exhibited clear limitations regarding dynamism, adaptivity, and semantic interpretability. Academic analyses of early API evolution emphasize that changes typically required extensive manual adjustments, making integration processes time-consuming and error-prone [35], [36].

3.2 Phase 2: AI-augmented API ecosystems

The second phase marks the transition toward AI-assisted API landscapes, where machine learning is applied to optimize traditional processes. Key developments include:

- Automatically generated documentation, often based on code analysis or log extraction, consistent with recent research on LLM-supported specification comprehension [31].
- Model-based test generation and continuous quality monitoring.
- Anomaly detection in API operations enabled through ML-based telemetry analysis, aligning with current work on OpenTelemetry-driven observability [13], [14].

APIs increasingly operate as data-driven system nodes whose behavior is dynamically analyzed, predicted, and optimized, evolving from static integration artifacts into active components of intelligent platform architectures.

3.3 Phase 3: AI-native API ecosystems

The current developmental phase is characterized by the emergence of AI-native API ecosystems—environments designed from the ground up for AI-driven interactions. In these settings, large language models (LLMs) and agentic systems do not merely consume API specifications; they can autonomously interpret, extend, modify, and orchestrate them. Recent empirical studies demonstrate that LLMs can internalize OpenAPI semantics, generate structurally consistent flows, and maintain schema adherence through constrained decoding [34].

Characteristic features of this phase include:

- Autonomous inference of missing structural elements in API specifications.
- Generation of semantically enriched interfaces.
- Multi-stage workflow orchestration without direct human control.
- Context-sensitive planning and dynamic parameter adaptation.
- Real-time enforcement of policies and governance rules by AI agents, reflecting modern agent architecture taxonomies [32], [33].

APIs shift from passive connectors to active interaction surfaces for autonomous systems that plan and execute complex task chains.

3.4 Technological convergence and architectural transformation

Parallel to functional evolution, the technical layers of API architectures are also changing. Alongside REST and GraphQL, AI-specific endpoints are emerging that:

- accept natural language as an interface,
- generate context-dependent responses,
- encode semantic interaction patterns.

This shift aligns with recent industry analyses predicting that over 30% of API demand growth by 2026 will originate from LLM- and AI-enabled systems [37]. Components such as model serving, prompt processing, and intelligent request routing lead to entirely new architectural patterns. Economically, APIs are transforming into AI-powered value-added services, whose usage increasingly depends on computational cost, model complexity, and adaptive pricing strategies.

3.5 Agentic AI and multi-agent ecosystems

A key development is the proliferation of agentic AI, giving rise to ecosystems in which multiple models or agents:

- cooperate,
- autonomously plan API call chains,
- jointly develop problem-solving strategies.

Research on multi-agent architectures highlights that such systems require new forms of coordination, safety, and performance measurement [32], [33]. This leads to increased demand for:

APIs have become critical infrastructure components that not only exchange data but increasingly mediate and control autonomous decision-making across interconnected agent systems.

4 AI-ENHANCED API FUNCTIONALITIES

AI-enhanced API functionalities fundamentally reshape how interfaces are interpreted, executed, and orchestrated. Modern large language models (LLMs) increasingly assume tasks that traditionally required human analytical reasoning—such as decision logic, schema comprehension, and multi-step integration planning—as documented in recent research on LLM function calling and agentic API reasoning [38], [39].

4.1 LLM-based function calling and semantic intent interpretation

LLM-based function calling enables the translation of natural-language instructions into machine-structured API operations. Studies show that LLMs autonomously infer parameters, validate inputs, complete missing fields, and design multi-step workflows. The AsyncLM system demonstrates that **asynchronous** function calling reduces end-to-end latency by **1.6× to 5.4×** and supports concurrent tool execution, significantly improving over synchronous approaches [40].

Fine-tuned models trained on OpenAPI-derived flows also exhibit strong structural adherence: they implicitly learn API constraints and outperform NER- and RAG-based baselines across both in-distribution and out-of-distribution scenarios [34].

Complementary research shows that specialized prompting formats, instruction-following data, and decision-token mechanisms further boost the reliability and multilingual robustness of LLM function-calling systems [41].

These advances support context-adaptive and semantically coherent API interactions in which LLMs dynamically tailor operations to user intent, system state, and domain constraints.

4.2 Semantic schema inference and structural completion

A defining capability of AI-native API ecosystems is semantic schema inference. Human-in-the-loop frameworks such as *LLMs4SchemaDiscovery* show that LLMs can extract and refine complex schema structures from unstructured text, subsequently grounding them in domain ontologies through agent-based alignment workflows [42], [43].

Schema matching experiments indicate that LLMs accurately detect semantic correspondences between schema attributes using names and descriptions alone, surpassing similarity-based baselines and reducing manual verification overhead [44].

Further evidence comes from schema-inference research on heterogeneous tabular datasets. SI-LLM, for example, can derive hierarchical type structures, attributes, and relationships using only column headers and sample values, yielding results competitive with or superior to traditional schema-learning techniques [45].

These developments transform APIs from static syntactic artifacts into semantic, machine-interpretable, and self-healing interfaces capable of detecting and correcting inconsistencies.

4.3 Adaptive and context-dependent interface behaviors

Adaptive APIs dynamically adjust parameters, field granularity, and response structures based on contextual signals such as authorization roles, governance policies, system load, and deployment environment. Research on human-in-the-loop evaluation frameworks for LLM-integrated applications emphasizes the necessity of context-sensitive and adaptive testing processes to ensure reliability across varied operational conditions [46].

As a result, modern API ecosystems support differentiated behavior across human users, autonomous agents, and distributed backend systems, extending classical API interaction models.

4.4 Generative endpoints and intelligent middleware functionalities

Generative endpoints blend traditional deterministic operations with LLM-powered reasoning, enabling APIs to generate semantically enriched, context-dependent, and dynamically transformed outputs.

Simultaneously, middleware systems are becoming increasingly intelligent. Research on AI-augmented middleware demonstrates how integration layers can proactively analyze signals, predict failures, optimize routing, and orchestrate distributed workflows [47].

Key intelligent capabilities include:

- Model selection and routing based on task profile or governance constraints, aligning with modern AI gateway architectures [48]
- Context expansion via retrieval and memory augmentation
- Predictive caching and response optimization
- Real-time policy enforcement, as highlighted in large-scale AI governance studies [49].

Through this shift, APIs become intelligent interaction layers—shaped by context, guided by policy, and integrated into ecosystems that learn and adapt.

5 AUTONOMOUS AGENTS AND API INTERACTION

Autonomous agents fundamentally expand API ecosystems by shifting from merely consuming interfaces to actively discovering, interpreting, and coordinating them. Recent research shows that large language models (LLMs) can operate as autonomous agents capable of interpreting instructions, reasoning over specifications, and managing sequential tool-enabled tasks [50]. These agents automatically analyze API specifications, derive internal representations of tools or functions, and autonomously plan multi-step workflows across multiple endpoints. Central to this capability are semantic understanding of interface logic and dynamic contextualization, both of which support situationally appropriate decision-making [4], [50], [51].

5.1 Automatic derivation and use of internal tool models

Modern agent architectures detect available APIs, extract their properties, and derive internal tool models that encode parameters, preconditions, states, and expected effects. LLM-based agents increasingly use structured tool-calling pipelines that allow them not only to follow static documentation but to act adaptively based on environmental feedback [50]. Research on tool-using LLM agents highlights their ability to select alternative execution paths, self-correct failure states, and iteratively optimize processes through feedback loops, improving both stability and efficiency [50], [51] [52], [53].

5.2 Coordination in multi-agent systems

A major evolution in this field is the development of multi-agent systems (MAS), in which multiple specialized agents collaborate via coordinated communication protocols and shared context structures. Surveys of multi-agent LLM systems emphasize the growing importance of agent-to-agent negotiation, collective deliberation, and distributed planning to enable complex problem solving [52]. Multi-agent research also identifies challenges in allocating tasks, managing layered context, and maintaining consistency and safety across agents with different roles [52]. Such systems allow cooperative problem solving, flexible scaling, and robust distribution of complex tasks [54], [55].

5.3 Security and governance challenges

As autonomy increases, new safety and governance challenges arise. Agentic systems introduce risks such as unintended or unauthorized API actions and complex failure modes. Security research shows that agentic AI systems require layered governance structures to mitigate threats, including adversarial manipulation, coordination failures, and misuse of tool-access capabilities [56]. Scholars emphasize safeguards such as whitelisting, controlled tool permissions, human-in-the-loop verification for high-risk actions, isolated execution environments, and post-hoc validation of results and side effects [56]. Risk assessments of multi-agent autonomous systems highlight that system trustworthiness depends not only on classic security properties but also on preserving behavioral alignment and preventing cascading failures across agent layers [56], [54], [56].

Different agentic frameworks—whether deterministic, deliberative, or message-oriented—exhibit trade-offs in speed, control, and safety. The choice of architectural model therefore strongly influences resilience and trustworthiness in API-centric AI systems [52], [56], [57].

6 SECURITY: CHALLENGES AND OPPORTUNITIES

The integration of AI-based components—particularly large language models (LLMs)—into API ecosystems introduces an expanded and qualitatively new class of security risks. Unlike deterministic rule-based systems, LLMs behave context-sensitively, making them vulnerable to manipulated inputs that can trigger unexpected or harmful behavior. Recent studies show that agentic and multi-agent LLM systems amplify these risks by enabling autonomous API actions that propagate through entire toolchains and execution pipelines, creating system-wide vulnerabilities [23], [24], [56].

6.1 Prompt injection attacks

Prompt injection represents one of the most severe threats to AI-augmented API systems. Research demonstrates that maliciously crafted inputs can induce LLMs to ignore safety constraints, execute unauthorized operations, leak sensitive information, or even override internal agent logic [23], [58], [59]. In multi-agent environments, the threat escalates significantly: compromised prompts can propagate between agents in a viral manner, triggering cascading API calls and multi-step exploit chains that bypass conventional protection mechanisms [23], [24].

6.2 Tool and API poisoning

Tool poisoning attacks target not the model itself, but the schemas, metadata, or tool representations used by agents to interact with APIs. Academic analyses highlight that subtle manipulation of tool specifications can cause agents to misinterpret parameters, violate preconditions, skip validation steps, or execute unintended actions [58], [60]. Because many agentic systems dynamically generate internal tool models at runtime, even small deviations can significantly alter system behavior and reliability [59], [60].

6.3 Cross-API exploit chains

An additional threat category arises from cross-API exploit chains, where attackers combine vulnerabilities across multiple APIs to construct complex multi-stage attack paths. Studies on agentic AI threat modeling emphasize that autonomous agents—especially those that independently plan API sequences—are highly susceptible when validation pipelines are insufficient, execution environments are weakly isolated, or telemetry is not centrally correlated [56], [60], [61]. Such chained exploits often evade traditional security systems, which fail to recognize multi-API cascades as a unified attack.

6.4 Defense-in-depth approaches

Mitigating these threats requires a layered defense strategy. Core defensive measures identified in recent security frameworks include strict input validation, context sanitization, isolation of agent memory and execution contexts, whitelisting of tools and permissible API operations, post-execution validation of critical actions, and continuous telemetry monitoring using metrics, traces, and logs [58], [61], [62]. Modern trust models further strengthen security with mechanisms such as mTLS for end-to-end trust, OAuth 2.1 for fine-grained authorization, and zero-trust architectures that validate each request independently of source or context [61], [62].

6.5 AI-supported defensive mechanisms

As system complexity increases, AI-driven defense mechanisms gain importance. Machine-learning-based anomaly detection, adaptive throttling, risk-scoring models that evaluate agent or client behavior, and behavior-based policy-enforcement engines allow real-time adaptation to evolving attack patterns. Academic work highlights that these techniques enable earlier detection of atypical patterns and improve resilience against novel or emerging adversarial techniques, particularly in autonomous and agentic AI environments [56], [62], [63].

7 GOVERNANCE, COMPLIANCE, AND ETHICS

Governance frameworks are becoming increasingly important in AI-enabled API ecosystems, as autonomous and semi-autonomous systems must comply not only with technical but also regulatory, organizational, and ethical requirements. With the proliferation of agentic AI, the need to ensure transparency, traceability, accountability, and control across the entire lifecycle of APIs, models, and interactions is rising significantly [16], [64].

7.1 Regulatory foundations: EU AI act

The EU AI Act (Regulation (EU) 2024/1689) establishes a risk-based regulatory model that is directly relevant for AI-supported API interactions. Recent academic analyses emphasize the Act’s proactive, rights-preserving orientation and its implications for autonomous decision-making systems [16], [65].

High-risk systems are subject to stringent requirements, including:

- comprehensive technical documentation and data-quality evidence [64],
- transparency obligations regarding system logic and decision-making grounds [16],
- continuous logging and monitoring mechanisms [64],
- human oversight (“human-in-the-loop/on-the-loop”) [16], [64], [65].

In agentic API ecosystems, these requirements apply particularly to automated decision processes, adaptive interfaces, and model-driven orchestration logic that may surpass regulatory thresholds.

7.2 AI management systems: ISO/IEC 42001:2023

ISO/IEC 42001:2023 defines an Artificial Intelligence Management System (AIMS) that helps organizations establish robust governance structures for AI. Academic studies show that the standard provides essential mechanisms for ensuring transparent, fair, unbiased, and secure AI operations [66], [67].

Key requirements include:

- clearly defined roles and accountability structures [66],
- risk-based assessment and control mechanisms [67],
- audit and verification procedures [66],
- policies for documentation, monitoring, and traceability of AI components [67].

For API ecosystems, this means that model and API versions, evaluation methods, test protocols, and interaction paths must be documented comprehensively and be fully manageable in accordance with AIMS specifications [66], [67].

7.3 Practical risk management: NIST AI RMF

The NIST AI Risk Management Framework (AI RMF) adopts a strongly practice-oriented approach and is structured into four core functions:

- MAP — identification and contextualization of risks,
- MEASURE — collection, analysis, and evaluation of metrics and risks,
- MANAGE — implementation of effective mitigation and control measures,
- GOVERN — establishment of organization-wide responsibilities and governance structures.

The RMF supports consistent risk detection and management across the lifecycle of models, APIs, and agentic interactions. Recent research emphasizes the importance of transparency requirements, documentation of risk and quality metrics, and ensuring human intervention capabilities, particularly for generative and autonomous AI deployments [68], [69].

7.4 Ethics: fairness, responsibility, and control of autonomous systems

As the autonomy of AI agents increases, ethical considerations become more prominent. Autonomous systems now make decisions that were previously reserved exclusively for humans, leading to growing expectations regarding:

- fairness and non-discrimination [70], [71],
- reproducibility of decisions [70],
- data ethics and principles of data minimization [71],
- auditability and explainability of decision processes [70],
- robust control mechanisms that detect and correct erroneous decisions early [71].

Research on ethical AI in autonomous systems underscores the importance of explainability, bias mitigation, stakeholder engagement, and accountability frameworks to ensure responsible integration and governance [70], [71]. Agentic systems must therefore be designed to maintain human oversight at all times, enabling immediate intervention—especially in high-critical API interactions [69], [70]. Moreover, the three major governance frameworks discussed above—the EU AI Act, ISO/IEC 42001:2023, and the NIST AI RMF—offer complementary yet distinct approaches to managing AI-related risks. Together, they provide a multilayered foundation for responsible AI operations, regulatory and organizational compliance, and the embedding of human-centric controls, as summarized in the crosswalk table below (*Table 2*).

Table 2

Governance Crosswalk (Source: Own Representation)

Governance Dimension	EU AI Act	ISO/IEC 42001:2023	NIST AI RMF
Scope / Purpose	Legally binding risk-based regulation for AI systems	Organizational AI Management System (AIMS) for responsible AI operations	Voluntary risk management guidance for trustworthy AI
Risk Classification	Explicit risk tiers (unacceptable, high-risk,	Requires risk assessment processes	MAP + MEASURE steps identify and

	limited-risk)	but does not define legal tiers	quantify risks dynamically
Documentation Requirements	Mandatory technical documentation, data quality evidence	Systematic documentation of models, APIs, versions, testing, and monitoring	Documentation recommended as part of risk measurement and governance
Transparency & Explainability	High-risk systems must disclose logic, capabilities, and limitations	Requires transparency policies for AI components	Transparency required through risk characterization and governance
Monitoring & Logging	Continuous monitoring and logging obligations	Ongoing monitoring and auditability processes	MEASURE + MANAGE require tracking performance, risks, and failures
Human Oversight	Mandatory for high-risk systems (“human-in/on-the-loop”)	Requires governance roles ensuring human oversight	Human intervention embedded in Govern/Manage functions
Ethics & Fairness	Focus on rights protection, non-discrimination	Ethical governance embedded in AIMS policies	Fairness and harm-mitigation considered core risk categories
Operational Integration	Applies to providers, importers, deployers	Applies to organizations implementing AI solutions	Applies across AI lifecycle and organizational contexts
Compliance Mechanism	Regulatory enforcement, penalties for non-compliance	Certification and internal audit	Self-assessment and alignment with best practices

8 ARCHITECTURE AND INTEROPERABILITY

Modern API architectures are evolving from statically defined interfaces toward context-sensitive, semantically enriched, AI-supported interaction models. This shift is essential for autonomous systems that not only consume but also interpret, augment, and dynamically orchestrate API specifications [72], [73].

8.1 Extended interface specifications: OpenAPI 3.2

OpenAPI 3.2 represents a significant advancement in machine-interpretable interface description standards. The specification introduces:

- hierarchical tagging,
- streaming media types for continuous data flows,
- additional operation types to model interactions more flexibly,

while maintaining full backward compatibility with OpenAPI 3.1 [72], [74].

These enhancements simplify the structuring of complex API landscapes and help AI models better understand semantic dependencies between endpoints—an essential advantage for autonomous agents and LLM-driven workflows [72], [74].

8.2 gRPC over HTTP/3: efficiency in distributed systems

The adoption of gRPC over HTTP/3 (QUIC) marks another milestone. Transitioning from HTTP/2 to HTTP/3 enhances:

- latency,
- connection stability,
- multiplexing efficiency,
- tolerance to packet loss,
- especially in distributed environments, mobile networks, and data-intensive architectures [75], [76].

Research shows QUIC reduces connection setup time, avoids head-of-line blocking, and supports seamless connection migration—benefits that are crucial for high-frequency AI-based API interactions [12], [76]. Experimental evaluations demonstrate significant latency and throughput improvements in gRPC deployments when HTTP/3 is used, making it increasingly considered best practice [75], [76].

8.3 OpenTelemetry: observability for AI-native API systems

As API interactions become increasingly automated, the demand for comprehensive observability rises sharply. OpenTelemetry is gaining importance in AI-native architectures by standardizing:

- tracing,
- metrics,
- logging,

across organizational and infrastructural boundaries [77], [78].

In combination with AI-augmented endpoints, OpenTelemetry enables detailed analysis of:

- model decisions,
- prompt chains,
- agent interactions,
- routing paths and orchestration logic,

which is vital for auditability, fault tolerance, risk assessment, and regulatory compliance [77], [79].

8.4 Architectural patterns and interoperability in AI ecosystems

AI-enabled API systems employ various architectural paradigms, including:

- microservices for modular and scalable services,
- serverless models for event-driven execution,
- specialized model-serving platforms supporting LLMs and ML pipelines [80], [81].

A critical challenge arises from ensuring interoperability among heterogeneous models, such as:

- closed-weight proprietary models,
- open-weight transparent models,
- multi-model architectures like Mixture-of-Experts systems [80], [82].

AI- and agent-based systems require consistent standards for:

- cross-model orchestration,
- error handling and rollbacks,
- policy enforcement,
- versioning of semantic interfaces [81], [82].

Only with such interoperability and governance mechanisms can AI-driven API ecosystems remain reliable, scalable, and compliant with regulatory expectations [80], [81], [82].

9 EFFICIENCY, PERFORMANCE, AND SUSTAINABILITY

The increasing integration of AI—particularly large language model (LLM)-based components—into API ecosystems results in substantially higher demands for computational capacity, latency optimization, and energy efficiency. Model inference, agent-based orchestration, and complex API workflow chains cause elevated resource consumption, making systematic optimization across infrastructure, model, and application layers indispensable [72], [74].

9.1 Model inference: energy and latency intensity of large language models

LLM inference remains one of the most resource-intensive processes in modern AI systems. Overall energy demand grows with factors such as:

- model size (number of parameters),
- context length (prompt and token window),
- parallel request execution, especially in multi-agent environments.

Efficiency-enhancing techniques—including model distillation, quantization, and structured compression—reduce computational load by providing smaller, optimized model variants while preserving key functional capabilities. These techniques significantly improve ecological footprint and scalability in agentic systems [72], [73].

Moreover, observability frameworks such as OpenTelemetry make inference performance bottlenecks and inefficiencies visible through granular traces and metrics, which supports data-driven optimization of energy-intensive AI operations [77], [78], [79].

9.2 Edge inference and caching: reducing latency and energy consumption

Edge inference is gaining importance for API-driven agent systems that must:

- make rapid decisions,
- meet low-latency requirements,
- or continuously process sensor data.

Local model execution substantially reduces network latency and data-transfer overhead. Complementary caching strategies—such as reusing previous computation paths or storing prompt representations—further reduce redundant model calls and lower the energy cost per request. These caching patterns are aligned with efficiency-focused microservice research showing that localized computation improves responsiveness and resource efficiency in distributed environments [80], [81].

9.3 Orchestration in multi-agent systems

Multi-agent systems increase architectural complexity and have a significant impact on overall energy usage. Major inefficiencies stem from:

- repeated or overlapping model calls,
- redundant planning steps,
- extended or non-optimized API call chains.

Energy-optimized orchestration strategies include:

- prioritization of agent chains to activate only relevant models,
- routing schemes that minimize computational load,
- reduction of redundant interactions through cached intermediate results,
- dynamic load distribution across models.

Studies on AI-supported microservices show that optimized orchestration and adaptive system behavior reduce computational overhead, latency, and energy consumption while enhancing operational robustness [81], [82].

9.4 Transport protocols: efficiency gains through HTTP/3

Modern transport protocols contribute significantly to performance and energy optimization.

HTTP/3, built on QUIC, offers:

- reduced latency via faster handshakes,
- robust multiplexing without head-of-line blocking,
- higher stability under packet loss,

- more efficient connections in unstable or variable network conditions [12], [75], [76].

Empirical evaluations of QUIC-based gRPC communication demonstrate measurable improvements in throughput and latency compared to HTTP/2, supporting more energy-efficient communication patterns for AI-heavy architectures that depend on high-frequency API calls [12], [76].

9.5 OpenTelemetry: data-driven optimization of efficiency and sustainability

OpenTelemetry provides deep, system-wide observability by capturing:

- performance data (latency, throughput, error rates),
- model invocation and inference patterns,
- agent interactions and routing decisions,
- cross-system resource utilization paths [77], [78], [79].

This telemetry supports data-driven optimization strategies such as:

- dynamic scaling of models,
- identification of inefficient API sequences,
- energy profiling of agents or workflows,
- automated recommendations for more resource-efficient execution paths.

Due to its standardized tracing, metrics, and logging capabilities, OpenTelemetry has become a foundational component in building sustainable and performance-optimized AI-driven API ecosystems [78], [79].

10 CROSS-DOMAIN APPLICATIONS AND CASE STUDIES

AI-enabled API ecosystems are increasingly widespread across societal and industrial domains. Autonomous API interactions and agent-driven workflows unlock substantial efficiency gains but simultaneously introduce new requirements regarding security, regulation, transparency, and accountability. The analysis of the public sector, healthcare, financial services, and mobility demonstrates domain-specific opportunities and risks [83], [84].

10.1 Public sector: compliance, transparency, and auditability

In the public sector, compliance, traceability, and rule-conformant execution are paramount. Agent-based systems increasingly orchestrate automated workflows such as:

- approval workflows,
- document classification and verification,
- rule-based decision processes in administrative environments.

Research on AI adoption in government underscores that transparency, auditability, interoperability, and strong governance frameworks are essential to prevent systemic risks and maintain public trust [83], [84]. Strict regulatory requirements—particularly the EU AI Act—necessitate comprehensive audit trails, robust policy controls, and transparent decision-making mechanisms.

Additionally, safety concerns such as prompt injection or erroneous model reasoning can directly affect administrative decisions, reinforcing the need for rigorous validation, continuous monitoring, and mandatory human oversight [84], [85].

10.2 Healthcare: safety, privacy, and clinical validity

AI-enabled API endpoints are increasingly used in healthcare for:

- diagnostic decision support,
- analysis of medical imaging and sensor data,
- patient-context interpretation in clinical information systems.

However, healthcare-specific risks are substantial. Empirical analyses show that LLM-based systems introduce new failure modes such as hallucinations, schema manipulation, or unsafe content generation, which may compromise patient safety [86], [87].

Healthcare researchers emphasize that unsafe or unvalidated AI interactions—especially through APIs that connect to electronic health records or diagnostic tools—can result in data-privacy breaches, clinical misinterpretations, or patient harm [86], [87], [88]. Consequently, these systems require:

- strict multi-layer validation,

- role-based access control,
- model- and API-specific audit trails,
- continuous monitoring of sensitive data flows [87], [88].

10.3 Financial sector: transparency, risk management, and regulatory control

In finance, AI-based API interactions are becoming integral for:

- regulatory reporting,
- fraud detection,
- risk assessments and compliance analytics.

Academic research on AI-driven financial risk management highlights that multi-agent architectures can identify complex transactional patterns, coordinate verification pathways, and correlate events across distributed services [89], [90].

Regulators increasingly mandate transparency, explainability, and reproducibility of model-driven decisions. Studies show that financial risk-management frameworks require:

- transparent model behavior,
- traceability of API decision flows,
- strong auditability guarantees,
- clearly defined human accountability in autonomous agent deployments [89], [90], [91].

The financial sector therefore remains one of the most tightly regulated areas for AI governance due to its systemic importance.

10.4 Mobility: real-time performance, safety, and energy efficiency

In the mobility sector—especially autonomous and semi-autonomous systems—APIs constitute the backbone for:

- real-time communication,
- sensor fusion,
- situational routing,

- vehicle and traffic coordination.

Recent research in autonomous mobility shows that agent-based AI systems must coordinate sensor streams, control logic, and environmental models at millisecond timescales, requiring continuous safety validation and robust failover designs [92], [93].

The integration of LLM-based decision modules in autonomous driving introduces new safety challenges related to real-time reasoning, interpretability, and regulatory compliance. Studies highlight the need for:

- strong safety validation frameworks,
- secure policy enforcement,
- energy-efficient model execution at the edge,
- consistent interoperability across vehicle, infrastructure, and cloud APIs [92], [93], [94].

Due to the direct physical consequences of incorrect or manipulated API interactions, mobility remains one of the most sensitive AI deployment domains.

Across all examined domains, AI-enabled API ecosystems provide significant potential for efficiency, automation, and data-driven decision-making. However, a trustworthy and safe deployment can only be ensured when:

- strict security mechanisms,
- comprehensive audit and logging procedures,
- domain-specific governance structures,
- and mandatory human oversight
- are consistently implemented.

11 TAXONOMY OF AI-DRIVEN API ECOSYSTEMS

This section introduces a multidimensional taxonomy that enables systematic classification, comparison, and capability assessment of AI-driven API ecosystems (*Table 3*). The taxonomy comprises five central dimensions that determine how effectively, securely, interoperably, and sustainably APIs can be deployed within agent-based systems. These dimensions reflect insights from contemporary research on AI governance, interoperability, predictive security, microservice automation, and autonomous decision-making [83], [84], [86], [92], [93], [95].

11.1 Degree of autonomy

The degree of autonomy describes a continuum from human-driven interactions to agent-guided and fully autonomous API orchestration:

- Human-guided: Decisions and API usage are executed explicitly by human users or developers.
- Agent-guided: Agents generate recommendations, plan interaction paths, and support human decision processes—similar to emerging agentic orchestration models discussed in AI governance research [83], [84].
- Fully autonomous: Systems independently orchestrate API chains, make decisions without human intervention, and adapt strategies contextually, reflecting autonomous AI system behaviors studied in mobility, finance, and microservices automation [92], [93].

This dimension reflects the progressive delegation of cognitive and operational tasks to autonomous agents, a trend consistently highlighted across sectors adopting AI-based automation [83], [86].

11.2 Interoperability

Interoperability is assessed along a spectrum:

- Syntactic interoperability: Compatibility at schema, structural, and protocol level—essential for basic cross-service communication.
- Semantic interoperability: Meaning-aligned representations of entities and relations, vital for consistent contextual understanding in AI systems [95].
- Adaptive interoperability: Context-aware adjustment of parameters, workflows, and data, enabling dynamic interpretation and autonomous decision-making in complex ecosystems [84].

Research in cross-domain AI governance and healthcare AI emphasizes that semantically adaptive interoperability is crucial for safe and explainable AI behavior at scale [86], [95].

11.3 Governance maturity

Governance maturity reflects the extent of organizational and technical control over AI-driven API interactions:

- Manual governance: Human-directed policy reviews and documentation, currently common in early-stage public sector AI deployments [83].
- Policy-driven governance: Automated policy checks, access controls, and compliance mechanisms—aligned with modern AI risk-management frameworks [84], [86].
- Self-enforcing governance: AI-supported evaluation, automated revisions, and continuous auditing—a model consistent with advanced autonomous system governance proposed in emerging research [93], [95].

Highly mature governance is required to detect risks early, maintain compliance, and ensure transparency in agentic systems [83], [84].

11.4 Security level

Security maturity progresses across:

- Static security: Traditional validation, schema checking, and authentication.
- Behavior-based security: Pattern recognition, anomaly monitoring, and telemetry-based analysis—the latter strongly recommended in healthcare and public sector AI deployments to detect emergent risks [86], [92].
- Predictive security: Proactive threat detection using risk analytics, historical usage patterns, and model-driven predictions. Studies highlight such predictive models as essential for mitigating LLM-specific vulnerabilities and preventing cascading system failures [86], [93].

With rising autonomy, predictive security becomes indispensable for addressing complex attack surfaces and high-impact risk scenarios [92], [93].

11.5 Energy efficiency

Energy efficiency reflects the shift from resource-intensive AI operations toward sustainable execution:

- Compute-intensive systems: Large models, long context windows, high concurrency—widely observed in modern multi-agent and LLM-driven pipelines [92].
- Energy-adaptive systems: Distillation, compression, dynamic context reduction—approaches validated in current AI optimization research for reducing computational overhead [93].
- Inference-balanced systems: Optimized load distribution, context-sensitive model routing, and energy-aware execution paths—critical for large-scale autonomous architectures and mobility systems [92], [93].

Energy efficiency is increasingly central to achieving scalable and sustainable multi-agent operations, especially in latency-sensitive domains such as healthcare and autonomous mobility [86], [92].

The table below presents the taxonomy in a multidimensional visual form, providing a clear and intuitive basis for evaluating how API ecosystems diverge in their levels of complexity, governance maturity, risk exposure, and suitability across domain-specific application contexts.

Table 3

A Systematic Taxonomy for Intelligent API Ecosystems (Source: Own Representation)

Dimension	Description	Maturity / Levels
1. Degree of Autonomy	Extent of autonomous decision-making and interaction	• Human-guided • Agent-guided • Fully autonomous
2. Interoperability	Structural, semantic, and adaptive API interaction	• Syntactic • Semantic • Adaptive / Context-sensitive
3. Governance Maturity	Organizational and technical control mechanisms	• Manual • Policy-driven • Self-enforcing
4. Security Level	Quality and maturity of security models	• Static • Behavior-based • Predictive
5. Energy Efficiency	Efficiency of model/API execution w.r.t. resource consumption	• Compute-intensive • Energy-adaptive • Inference-balanced

12 RESEARCH GAPS AND FUTURE WORK

Open Despite substantial progress in AI-enabled API ecosystems, several critical research gaps remain that must be addressed to ensure the secure, scalable, and trustworthy deployment of agentic systems. These challenges span foundational technical issues as well as governance, security, and sustainability concerns.

12.1 Lack of security benchmarks and test frameworks

A major unresolved issue concerns the absence of standardized security benchmarks for interactions between autonomous agents and APIs. Current research in multi-agent safety and verification indicates that the complexity of distributed agent behaviors requires robust evaluation frameworks, yet such frameworks remain underdeveloped. Recent advances in probabilistic and resource-bounded verification demonstrate the importance of formalized assessment tools for identifying vulnerabilities in multi-agent decision-making under uncertainty, highlighting the need for reproducible datasets, attack scenarios, and robustness metrics [96], [97].

12.2 Insufficient development of semantic interoperability

While syntactic interoperability (e.g., OpenAPI, gRPC) is well established, semantic interoperability remains an open research challenge. The academic literature emphasizes that interoperable systems must rely on formal, machine-interpretable semantic models enabling agents to consistently interpret data across heterogeneous environments. Recent work in semantic interoperability for IoT and knowledge-driven systems stresses the necessity of domain-spanning vocabularies, ontologies, semantic constraints, and graph-based knowledge structures to ensure consistent interpretation and reasoning [98], [99].

Although early initiatives demonstrate the potential of ontologies to bridge heterogeneous models, an integrated and standardized semantic framework for large-scale agentic API ecosystems is still lacking.

12.3 Underdeveloped methods for auditing autonomous API interactions

As agentic systems increasingly generate dynamic API interaction chains, the ability to audit intentions, decisions, and execution paths becomes essential. Academic research on accountability and explainability in autonomous systems highlights the importance of immutable logging, cryptographically secured evidence trails, and structured representations of agent rationales for ensuring compliance and transparency [100].

Existing methods, however, do not yet offer standardized representations for reconstructing agent decision paths or verifying execution integrity across distributed systems. This lack poses compliance challenges for emerging regulatory frameworks such as the EU AI Act and ISO/IEC 42001.

12.4 Limited understanding of energy and performance profiles in complex multi-agent systems

Although initial studies indicate that multi-agent LLM-based orchestration may impose substantial computational overhead, systematic academic analyses remain scarce. Recent work on energy efficiency in multi-agent LLM systems demonstrates measurable gains when employing energy-aware scheduling and adaptive collaboration mechanisms, yet also confirms the need for formal evaluation methodologies for LLM inference, latency profiles, caching strategies, and inter-agent parallelism [101].

A scientifically grounded assessment of energy consumption is critical for building sustainable AI ecosystems; current knowledge remains fragmentary.

12.5 Missing interoperable protocols and formal safety guarantees

Classical APIs operate deterministically, but agentic systems employ probabilistic, context-dependent decision processes. Academic work in formal verification of probabilistic multi-agent systems shows that ensuring predictable and safe behavior in such systems requires advanced mathematical frameworks capable of expressing and verifying probabilistic strategies, resource-bounded behaviors, and safety constraints [96], [102].

However, no interoperable security protocols or formal safety models yet exist that can guarantee mathematically verifiable safety for autonomous agentic APIs. Developing verifiable control models, fail-safe mechanisms, and isolation strategies remains a central research direction.

The identified research gaps—including missing security benchmarks, limited semantic interoperability, insufficient auditability, inadequate energy profiling, and absent formal safety guarantees—demonstrate that AI-driven API ecosystems are still in the early stages of scientific maturation. Ensuring their secure and sustainable deployment will require coordinated research initiatives, multidisciplinary approaches, and the development of new standards and reference architectures grounded in robust academic methodologies.

13 CONCLUSION

API ecosystems are evolving—driven by AI-powered automation—from static integration interfaces into highly dynamic and intelligent interaction layers. Increasingly, models and autonomous agents function as primary consumers and producers of API calls, positioning APIs as the operational foundation of modern intelligent systems. Ensuring the responsible scaling of these architectures requires robust orchestration mechanisms, comprehensive observability concepts, strict security controls, and alignment with regulatory frameworks such as the EU AI Act, ISO/IEC 42001, and the NIST AI Risk Management Framework. Recent academic research underscores that federated and AI-assisted observability models are essential to maintaining secure, privacy-preserving, and scalable operations across distributed cloud environments, while AI-augmented API management significantly enhances governance, anomaly detection, and automated policy enforcement within increasingly complex API landscapes.

With the rising autonomy of agentic systems, the demands for transparent decision pathways, explainable execution logic, and sustainable infrastructural models are growing. Studies on agent-centric API architectures emphasize that existing enterprise APIs—designed mainly for human-driven interactions—must adapt to flexible, goal-oriented, and context-aware communication patterns required by autonomous agents. Moreover, advances in security orchestration and automated response (SOAR) platforms demonstrate the need for agent-enabled, hyper-automated frameworks that ensure reliable, verifiable, and regulation-aligned execution of critical operational processes.

This work integrates established API architectural principles with current AI-based interaction patterns, presenting a comprehensive analysis of the functional, security-related, governance-oriented, and energy-focused dimensions of future API landscapes. Research on AI agent systems reveals that next-generation architectures depend on unified taxonomies covering agent reasoning, orchestration patterns, tool-use capabilities, and safety-aligned evaluation mechanisms. Additionally, academic studies on autonomous AI-infrastructure management highlight that self-managing agent systems play a crucial role in increasing system resilience, reducing operational overhead, and supporting sustainability-focused resource optimization.

The taxonomy developed here, together with the identified research gaps, provides a scientifically grounded foundation for the standardization, advancement, and responsible implementation of AI-driven API ecosystems.

ACKNOWLEDGMENTS

The author thanks Professor D. Atanasova for her support throughout this research.

ETHICS AND COMPLIANCE STATEMENT

This study adheres to PRISMA 2020 reporting; no human subjects or personal data were used. Recommendations align with EU AI Act and ISO/IEC 42001 [19], [67].

AUTHOR CONTRIBUTIONS

Conceptualization, Methodology, Investigation, Writing—Original Draft & Review: F. Témolé.

DATA AVAILABILITY STATEMENT

All references are publicly accessible; figures are programmatically generated and included.

FUNDING

The authors declare that no external funding was received.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] M. Posada and L. Vaccari, "APIs for Governments: why, what & how," European Commission JRC—APIdays Paris, https://interoperable-europe.ec.europa.eu/, 2020.

[2] S. -P. Ma, M. -J. Hsu, H. -J. Chen and C. -J. Lin, "RESTful API Analysis, Recommendation, and Client Code Retrieval," Electronics, vol. 12, no. 5, p. 1252, doi: 10.3390/electronics12051252, 2023.

[3] L. Wang et al., "A survey on large language model based autonomous agents," Frontiers of Computer Science, vol. 18, doi: 10.1007/s11704-024-40231-1, 2024.

[4] A. Yehudai et al., "Survey on Evaluation of LLM-based Agents," arXiv, 2503.16416, doi: 10.48550/arXiv.2503.16416, 2025.

[5] T. Schick et al., "Toolformer: Language Models Can Teach Themselves to Use Tools," arXiv, 2302.04761, doi: 10.48550/arXiv.2302.04761, 2023.

[6] M. Li et al., "API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs," EMNLP, pp. 3102–3116, doi: 10.18653/v1/2023.emnlp-main.187, 2023.

[7] X. Li, "Survey of LLM-based Agents: Theories, Technologies, Applications," IEEE (Early Access), 2025.

[8] J. S. dos Santos et al., "Analysis of Tools for REST Contract Specification in Swagger/OpenAPI," ICEIS, https://www.scitepress.org/Papers/2020/93812/93812.pdf, 2020.

[9] A. Lercher et al., "Generating Accurate OpenAPI Descriptions from Java Source Code," arXiv, 2410.23873, 2024.

[10] M. Niswar et al., "Performance Evaluation of Microservices Communication with REST, GraphQL, and gRPC," Int. J. Electronics and Telecommunications, vol. 70, no. 2, pp. 429–436, doi: 10.24425/ijet.2024.149562, 2024.

[11] N. Hamo and S. Saberian, "Evaluating the performance and usability of HTTP vs gRPC," BTH Thesis, https://www.diva-portal.org/smash/get/diva2:1768795/FULLTEXT02.pdf, 2023.

[12] I. Khan and M. K. Ahamad, "Enhancing Security and Performance of gRPC-Based Microservices using HTTP/3 and AES-256," Journal of Information Systems Engineering & Management, https://www.scitepress.org/Papers/2020/93812/93812.pdf, 2025.

[13] F. A. A. Gomes et al., "Impact of OpenTelemetry Configuration on Observability and Telemetry Storage Cost," ADVANCE Workshop, https://hal.science/hal-04723959v1/file/ADVANCE_2024_Almada_1_.pdf, 2024.

[14] E. Norgren, "Optimizing Distributed Tracing Overhead in a Cloud Environment with OpenTelemetry," Master’s Thesis, https://www.diva-portal.org/smash/get/diva2:1867119/FULLTEXT01.pdf, 2024.

[15] D. Gurumurthy and L. Querel, "OpenTelemetry Semantic Conventions and How to Avoid Broken Observability," USENIX SREcon25 Americas, 2025.

[16] A. Mahmutovic, "EU AI Act: a proactive framework for comprehensive AI regulation," Int’l J. of Law and Information Technology, doi: 10.1093/ijlit/eaaf028, 2025.

[17] S. Greenstein and M. Zamboni, "Navigating the legislative dilemma: evaluating the EU AI Act," Theory and Practice of Legislation, vol. 13, no. 3, doi: 10.1080/20508840.2025.2513177, 2025.

[18] H. Graux et al., "Interplay between the AI Act and the EU digital legislative framework," European Parliament Study, https://www.europarl.europa.eu/RegData/etudes/STUD/2025/778575/ECTI_STU%282025%29778575_EN.pdf, 2025.

[19] M. Seet, "ISO 42001 and Legal Compliance: A Principled Implementation of the AI Management System," Springer, https://link.springer.com/book/10.1007/979-8-8688-2099-1, 2025.

[20] K. L. Lucy, "Overview of ISO/IEC 42001," UNIDO/Microsoft Presentation, https://www.unido.org/sites/default/files/files/2025-07/Microsoft%20-%20Overview%20of%20ISO%20IEC%2042001.pdf, 2025.

[21] R. Dotan et al., "Evolving AI Risk Management: A Maturity Model based on NIST AI RMF," arXiv, 2401.15229, https://arxiv.org/pdf/2401.15229, 2024.

[22] R. M. S. Khan et al., "Agents Under Siege: Breaking Pragmatic Multi-Agent LLM Systems with Optimized Prompt Attacks," ACL, pp. 8743–8759, https://aclanthology.org/2025.acl-long.476.pdf, 2025.

[23] D. Lee and M. Tiwari, "PROMPT INFECTION: LLM-to-LLM Prompt Injection within Multi-Agent Systems," arXiv, 2410.07283, https://arxiv.org/pdf/2410.07283, 2024.

[24] S. Gulyamov et al., "Prompt Injection Attacks in LLMs and AI Agent Systems: A Comprehensive Review," Information (MDPI), vol. 17, no. 1, doi: 10.3390/info17010054, 2026.

[25] A. Patel, "Designing Enterprise-Grade Microservices Security," Implementing Security with AI in GCP, Springer, pp. 113–132. (mTLS & Istio service mesh), https://link.springer.com/chapter/10.1007/979-8-8688-2213-1_6, 2026.

[26] E. K. Kähler et al., "Modular Security Analysis of OAuth 2.0 in the Three-Party Setting," IEEE, https://ieeexplore.ieee.org/document/9230361, 2020.

[27] S. Rose, O. Borchert, S. Mitchell and S. Connelly, "Zero Trust Architecture," NIST SP 800-207, doi: 10.6028/NIST.SP.800-207, 2020.

[28] J. Viswanathan, D. Kumar. N and S. Udhaya Kumar, "Zero Trust Security for Web Applications in Microservice-Based Architectures," IEEE, https://ieeexplore.ieee.org/document/10960955, 2025.

[29] M. J. Page et al., "PRISMA 2020 explanation and elaboration," BMJ, 372:n160, doi: 10.1136/bmj.n160, 2021.

[30] M. J. Page et al., "The PRISMA 2020 statement," BMJ, 372:n71, doi: 10.1136/bmj.n71, 2021.

[31] A. A. B. Aissa et al., "An LLM-Powered API Navigator: Building an Intelligent Assistant for API Specification Understanding," IEEE Feedforward Magazine, vol. 4, no. 3, pp. 1–15, https://hal.science/hal-05234168v1/file/AK_Beckn2025.pdf, 2025.

[32] B. Xu, "AI Agent Systems: Architectures, Applications, and Evaluation," arXiv:2601.01743, https://arxiv.org/abs/2601.01743, 2026.

[33] S. Sajjadi et al., "A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications," lectronics, vol. 14, no. 18, https://www.mdpi.com/2079-9292/14/18/3580, 2025.

[34] R. Chan et al., "Adapting LLMs for Structured Natural Language API Integration," EMNLP Industry Track, pp. 991–1000, https://aclanthology.org/2024.emnlp-industry.74/, 2024.

[35] M. Lamothe et al., "A Systematic Review of API Evolution Literature," ACM Computing Surveys, https://users.encs.concordia.ca/~shang/pubs/mlamothe_csur_2021.pdf, 2020.

[36] F. Di Lauro et al., "Towards Large-scale Empirical Assessment of Web APIs Evolution," APIACE/ICWE Workshops, https://design.inf.usi.ch/sites/default/files/biblio/apiace-icwe2021-api-evolution.pdf, 2021.

[37] Gartner, "More Than 30% of the Increase in API Demand Will Come From AI and LLM Tools by 2026," Gartner Press Release, https://www.gartner.com/en/newsroom/press-releases/2024-03-20-gartner-predicts-more-than-30-percent-of-the-increase-in-demand-for-apis-will-come-from-ai-and-tools-using-llms-by-2026, 2024.

[38] W. Liu et al., "ToolACE: Winning the Points of LLM Function Calling," arXiv:2409.00920, https://arxiv.org/abs/2409.00920, 2024.

[39] D. Kim et al., "Beyond Perfect APIs: A Comprehensive Evaluation of LLM Agents Under Real-World API Complexity," arXiv:2601.00268, https://arxiv.org/pdf/2601.00268, 2026.

[40] I. Gim, S. Lee and L. Zhong, "Asynchronous LLM Function Calling," arXiv:2412.07017, https://arxiv.org/abs/2412.07017, 2024.

[41] Y. -C. Chen et al., "Enhancing Function-Calling Capabilities in LLMs," NAACL Industry Track, https://aclanthology.org/2025.naacl-industry.9.pdf, 2025.

[42] S. Sadruddin et al., "LLMs4SchemaDiscovery: A Human-in-the-Loop Workflow for Scientific Schema Mining," arXiv:2504.00752, https://arxiv.org/abs/2504.00752, 2025.

[43] S. Sadruddin et al., "SCHEMA-MINERpro: Agentic AI for Ontology Grounding," Semantic Web Journal, https://www.semantic-web-journal.net/system/files/swj3871.pdf, 2025.

[44] M. Parciak et al., "Schema Matching with Large Language Models," VLDB TaDA Workshop, https://tabular-data-analysis.github.io/tada2024/papers/TaDA.8.pdf, 2024.

[45] P. WoL et al., "Schema Inference for Tabular Data Repositories Using Large Language Models," arXiv:2509.04632, https://arxiv.org/abs/2509.04632, 2025.

[46] B. John et al., "Adaptive Human-in-the-Loop Testing for LLM-Integrated Applications,"

https://www.researchgate.net/publication/391908960_Adaptive_Human-in-the-Loop_Testing_for_LLM-Integrated_Applications, 2025.

[47] A. Ndlovu and I. Mahlangu, "AI-Augmented Middleware: A New Paradigm for Intelligent Enterprise Integration," https://www.researchgate.net/profile/Israel-Godwin-Mahlangu/publication/396684560_AI-Augmented_Middleware_A_New_Paradigm_for_Intelligent_Enterprise_Integration/links/68f5287cffdca73694b9010b/AI-Augmented-Middleware-A-New-Paradigm-for-Intelligent-Enterpris, 2025.

[48] Y. Lin, "AI Gateways: The Future Trend of AI Infrastructure," Apache APISIX, https://apisix.apache.org/blog/2025/06/18/ai-gateway-future-trend-of-ai-infrastructure/, 2025.

[49] A. Marshan, "AI-Augmented Teaching and Assessment in Higher Education," IEEE CAI Workshop, https://www.ieeesmc.org/cai-2026/w1-teaching/, 2026.

[50] S. S. Chowa et al., "From language to action: a review of large language models as autonomous agents and tool users," Artificial Intelligence Review, https://link.springer.com/article/10.1007/s10462-025-11471-9, 2026.

[51] X. Li et al., "A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges," Vicinagearth, vol. 1, article 9, https://link.springer.com/article/10.1007/s44336-024-00009-2, 2024.

[52] S. Han, Q. Zhang, Y. Yao, W. Jin and Z. Xu, "LLM Multi-Agent Systems: Challenges and Open Problems," arXiv preprint, arXiv:2402.03578, https://arxiv.org/abs/2402.03578, 2024/2025.

[53] Y. Gao and S. Wu, "A Four-Layer Security Governance Framework for LLM-Based AI Agents," Journal of Artificial Intelligence Practice, vol. 8, no. 4, https://www.clausiuspress.com/assets/default/article/2026/01/07/article_1767842801.pdf, 2025.

[54] S. Raza, R. Sapkota, M. Karkee and C. Emmanouilid, "TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Multi-Agent Systems," arXiv preprint, arXiv:2506.04133, https://arxiv.org/pdf/2506.04133, 2025.

[55] Y. Wu et al., "Multi-Agent Autonomous Driving Systems with Large Language Models," indings of ACL: EMNLP, pp. 12756–12773, https://aclanthology.org/2025.findings-emnlp.683/, 2025.

[56] M. Leo, F. Tan, T. Miao and G. Anand, "From threat to trust: assessing security risks of agentic AI systems," International Journal of Information Security, vol. 25, article 23, https://link.springer.com/article/10.1007/s10207-025-01185-y, 2026.

[57] K. Grimes et al., "SOK: Bridging Research and Practice in LLM Agent Security," Carnegie Mellon SEI, https://sei.cmu.edu/documents/6414/Bridging-Research-and-Practice-in-LLM-Agent-Security.pdf, 2025.

[58] OWASP, "LLM Prompt Injection Prevention Cheat Sheet," OWASP, https://cheatsheetseries.owasp.org/cheatsheets/LLM_Prompt_Injection_Prevention_Cheat_Sheet.html, 2025.

[59] B. Hofesh, "Prompt Injection vs. Data Poisoning: The Two Biggest Security Threats to LLM Applications," Bright Security, https://brightsec.com/blog/prompt-injection-vs-data-poisoning-the-two-biggest-security-threats-to-llm-applications/, 2025.

[60] V. S. Narajala, "Securing Agentic AI: A Comprehensive Threat Model and Mitigation Framework," arXiv, https://arxiv.org/pdf/2504.19956, 2025.

[61] T. Erlin, "The API Imperative: Securing Agentic AI and Beyond," Security Boulevard, https://securityboulevard.com/2025/04/the-api-imperative-securing-agentic-ai-and-beyond/, 2025.

[62] OWASP GenAI Project, "OWASP Top 10 for LLM Applications," OWASP , https://owasp.org/www-project-top-10-for-large-language-model-applications/, 2025.

[63] J. Braidwood, "LLM Security Guide: OWASP Top 10 & Defenses 2026," GLACIS, https://www.glacis.io/guide-llm-security, 2026.

[64] European Union Agency for Fundamental Right (FRA), "Assessing High-Risk Artificial Intelligence: Fundamental Rights Risks," https://fra.europa.eu/en/publication/2025/assessing-high-risk-ai, 2025.

[65] CMS LawNow, "2024 EU AI Act: A detailed analysis," https://cms-lawnow.com/en/ealerts/2025/03/2024-eu-ai-act-a-detailed-analysis, 2025.

[66] S. Biroğul, Ö. Şahin and H. Əsgərli, "Exploring the Impact of ISO/IEC 42001:2023 AI Management Standard on Organizational Practices," Advances in Artificial Intelligence Research, vol. 5, no. 1, pp. 14–22, https://dergipark.org.tr/en/pub/aair/issue/92433/1709628, 2025.

[67] ISO, "ISO/IEC 42001:2023," Artificial Intelligence Management System, https://www.iso.org/standard/42001, 2023.

[68] National Institute of Standards and Technology, "AI Risk Management Framework (AI RMF 1.0)," NIST, https://www.nist.gov/itl/ai-risk-management-framework, 2023.

[69] NIST, "Artificial Intelligence Risk Management Framework: Generative AI Profile (NIST AI 600-1)," NIST, https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf, 2024.

[70] O. O. Ajayi, A. S. Adebayo and N. Chukwurah, "Ethical AI and Autonomous Systems: A Review of Current Practices and a Framework for Responsible Integration," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 1, no. 1, https://www.futureengineeringjournal.com/uploads/archives/20250315123154_FEI-2025-1-003.1.pdf, 2024.

[71] A. Batool, D. Zowghi and M. Bano, "AI governance: a systematic literature review," AI and Ethics, vol. 5, https://link.springer.com/article/10.1007/s43681-024-00653-w, 2025.

[72] C. Wood, "Announcing OpenAPI v3.2," OpenAPI Initiative, https://www.openapis.org/blog/2025/09/23/announcing-openapi-v3-2, 2025.

[73] OAI/OpenAPI-Specification, "The OpenAPI Specification Repository," GitHub, https://github.com/OAI/OpenAPI-Specification, 2025.

[74] N. Park, "What's New in OpenAPI 3.2," Zylo Docs Blog, Aug, https://blog.zylosystems.com/posts/openapi-3-2-key-changes-for-api-documentation, 2025.

[75] M. Noël, "Resilient connectivity for gRPC using Multipath QUIC," École polytechnique de Louvain, https://thesis.dial.uclouvain.be/, 2024/2025.

[76] T. Dang, "gRPC over HTTP/3 in Production," ThinhDA Engineering Blog, https://thinhdanggroup.github.io/grpc-over-http3/, 2025.

[77] N. Gazit and G. Liu, "Observability for Large Language Models with OpenTelemetry," OSACon , https://osacon.io/slides/2024/Observability-for-Large-Language-Models-with-OpenTelemetry.pdf, 2024.

[78] Ashnik Team, "Insights From the 2024 Observability Landscape," Ashnik Insights, https://www.ashnik.com/, 2024.

[79] D. Hope, "The Next Evolution of Observability with OpenTelemetry and Generative AI," Elastic Observability Labs, https://www.elastic.co/, 2025.

[80] D. Narváez et al., "Designing Microservices Using AI: A Systematic Literature Review," Software, vol. 4, no. 1, https://www.mdpi.com/, 2025.

[81] J. Willard and J. Hutson, "The Evolution and Future of Microservices Architecture with AI-Driven Enhancements," IJRES, vol. 12, no.1, https://www.ijresonline.com/, 2025.

[82] B. Sanwouo, P. Temple and C. Quinton, "Generative AI-based Adaptation in Microservices Architectures," ICWS , https://hal.science/hal-05082732v1/file/ICWS%2725.pdf, 2025.

[83] A. Aarab et al., "Integrating AI in Public Governance: A Systematic Review," Digital, vol. 5, no. 4, https://www.mdpi.com/2673-6470/5/4/59, 2025.

[84] OECD, "Governing with Artificial Intelligence: Are Governments Ready?," OECD AI Papers No. 20, https://www.oecd.org/en/publications/governing-with-artificial-intelligence_26324bc2-en.html, 2024.

[85] S. Pulijala, "Artificial Intelligence in Governance: Opportunities, Challenges, and Ethical Implications," IJFMR, https://www.ijfmr.com/papers/2024/6/29990.pdf, 2024.

[86] X. Wang et al., "Safety Challenges of AI in Medicine in the Era of Large Language Models," arXiv, https://arxiv.org/abs/2409.18968, 2024/2025.

[87] J. C. L. Chow and K. Li, "Large Language Models in Medical Chatbots: Opportunities, Challenges, and AI Risks," Information, vol. 16, no. 7, https://www.mdpi.com/2078-2489/16/7/549, 2025.

[88] F. De Micco et al., "Artificial Intelligence in Healthcare: Transforming Patient Safety," Frontiers in Medicine, https://www.frontiersin.org/articles/10.3389/fmed.2024.1522554/full, 2025.

[89] S. Joshi, "Review of Gen AI Models for Financial Risk Management," IJISEM, vol. 4, no. 2, https://satyadharjoshi.com/, 2025.

[90] Z. Feng et al., "Leveraging Artificial Intelligence in Financial Risk Management," JFRM, vol. 14, no. 2, https://www.scirp.org/pdf/jfrm_2410975.pdf, 2025.

[91] S. T. Battula, "AI-Driven Risk Management for Fintech Enterprises," IJSAT, https://www.ijsat.org/papers/2025/1/2804.pdf, 2025.

[92] Y. Wang et al., "Generative AI for Autonomous Driving: Frontiers and Opportunities," arXiv, https://arxiv.org/abs/2505.08854, 2025.

[93] R. Acharya, "LLM Integration in Autonomous Vehicle Systems," WJARR, https://journalwjarr.com/, 2025.

[94] IEEE MOST 2025, "Call for Papers: Mobility, Autonomous Systems, and AI," IEEE Mobility Conference, https://ieeemobility.org/MOST2025/call_for_papers.php, 2025.

[95] A. Ghosh, A. Saini and H. Barad, "Artificial Intelligence in Governance: Recent Trends, Risks, Challenges, and Future Directions," AI & Society, vol. 40, https://link.springer.com/article/10.1007/s00146-025-02312-y, 2025.

[96] D. Kurpiewski et al., "Formal Verification of Probabilistic Multi-Agent Systems for Ballistic Rocket Flight Using Probabilistic Alternating-Time Temporal Logic," arXiv preprint, https://arxiv.org/abs/2511.22572, 2025.

[97] H. N. Nguyen and A. Rakib, "Formal Modelling and Verification of Probabilistic Resource Bounded Agents," Journal of Logic, Language and Information, vol. 32, pp. 829–859, https://link.springer.com/article/10.1007/s10849-023-09405-1, 2023.

[98] A. Cimmino, M. Poveda-Villalón and R. García-Cast, "Ontologies and Semantic Interoperability," Springer Handbook of Internet of Things, https://link.springer.com/chapter/10.1007/978-3-031-39650-2_17, 2023.

[99] M. Stäbler et al., "Why an Automated, Scalable and Resilient Service for Semantic Interoperability is Needed," Proceedings of AI Safety and Security Research, https://pdfs.semanticscholar.org/ad48/ddec71c54e8fbc91ed5cbdc8dd7d8677bdbd.pdf, 2023.

[100] L. Fernández-Becerra et al., "Enhancing Trust in Autonomous Agents: An Architecture for Accountability and Explainability," arXiv preprint, https://arxiv.org/pdf/2403.09567, 2024.

[101] P. K. Goel, S. P. Yadav and P. Upadhyay, "Sustainability in Multi-Agent LLM Systems: Energy Efficiency and Green AI Initiatives," Advancements in Multi-Agent Large Language Model Systems for Next-Generation AI, IGI Global, https://www.igi-global.com/chapter/sustainability-in-multi-agent-llm-system/389183, 2026.

[102] H. Wang, A. Papachristodoulou and K. Margellos, "Distributed Safe Control Design and Probabilistic Safety Verification for Multi-Agent Systems," Automatica, vol. 179, https://kostasmargellos.github.io/assets/downloads/publications/journals/AUT_WPM_2023.pdf, 2025.

Authors' Contribution

All authors contributed equally to the development of this article.

Data availability

All datasets relevant to this study's findings are fully available within the article.

How to cite this article (APA)

Témolé, F., & Atanasova, D. (2026). API ECOSYSTEMS IN THE AGE OF ARTIFICIAL INTELLIGENCE. *Veredas Do Direito*, 23(4), e234344. https://doi.org/10.18623/rvd.v23.n4.4344