Governing Delegated Authority in Agentic AI: Designing an Enterprise Governance Architecture

Faculty of Science and Engineering
University of Groningen

April 2026

Word count: 15.094

Author

Jaimy-Lee Buiting – *S3097641*

1st Supervisor
dr. N.B. Szirbik

2nd Supervisor
dr. ir. G.H. Jonker

Abstract

Agentic AI systems create a governance challenge that existing responsible AI approaches only partly address: governing what such systems are authorised to do in practice. Current frameworks provide principles, risk controls, and oversight mechanisms, but are less developed in specifying, monitoring, and revising delegated authority across the lifecycle of agentic AI deployments. This research addresses that gap by asking: *what enterprise governance architecture is needed to specify, monitor, and revise delegated authority in agentic AI systems across their lifecycle?*

The research applies Design Science Research as its overarching methodology. Problem investigation combined a literature review with an embedded case study at CGI Netherlands to identify governance gaps related to delegated authority. These findings informed the design of the Delegation Governance Architecture (DGA), an enterprise governance architecture comprising four core functions: specifying the authority boundary, determining delegated authority, defining monitoring and enforcement configuration, and calibrating delegated authority over time through review and revision. A versioned Delegation Decision Artefact (DDA) records these decisions and their evolution.

The artefact was evaluated through ex ante application, organisational-fit analysis, and practitioner validation. The results indicate that the DGA provides a coherent structure for making delegated authority explicit and reviewable, while also showing that its use depends on organisational prerequisites that are often only partly established in practice.

The research contributes both a conceptual clarification and a design artefact. It positions delegated authority as a distinct governance object in agentic AI and proposes an enterprise architecture for governing it.

Use of AI Tools

This research made use of AI-assisted tools to support language refinement and visual presentation. Google Gemini was used to refine the layout and clarity of figures. Anthropic Claude was used to assist with editing, structuring, and improving the clarity of written text. All conceptual content, analysis, and design decisions, including the development of the Delegation Governance Architecture (DGA), are the original work of the author.

Table of Contents

List of Tables

List of Figures

Abbreviations

A-AI agentic artificial intelligence.

AI artificial intelligence.

AI RMF AI Risk Management Framework.

DCS Director Consulting Services.

DDA Delegation Decision Artefact.

DGA Delegation Governance Architecture.

DSR Design Science Research.

EU AI Act Regulation (EU) 2024/1689 on Artificial Intelligence.

GenAI generative AI.

HITL human-in-the-loop.

IDEF0 Integrated DEFinition for Function Modelling.

ISO/IEC 42001 ISO/IEC 42001:2023 Artificial Intelligence Management System.

LLM large language model.

NIST National Institute of Standards and Technology.

RAI responsible AI.

SSM Soft Systems Methodology.

Chapter 1

Introduction

As of early 2026, artificial intelligence (AI) agents have moved beyond experimental prototypes to become a standard tool in the modern software stack. In many cases, these agents are deployed as components that support specific tasks, such as code generation, information retrieval, or workflow assistance. However, agentic artificial intelligence (A-AI) systems extend beyond such component-level use. They embed computational components within iterative execution loops that enable planning, tool use, interaction with external systems and environments, and feedback-driven adaptation [1]. In practice, this allows such systems to coordinate and execute actions over time rather than merely respond to isolated inputs, as is typical of earlier generative AI (GenAI) applications. Major technology providers have rolled out agents capable of browsing the web, navigating file systems, and managing multi-step workflows with limited human intervention [2, 3]. Open-source frameworks have simultaneously enabled individual developers to run local agents that communicate with databases and operating systems through standardised protocols [4]. GitHub now supports the concurrent execution of multiple specialised coding agents within a single development environment, allowing teams to delegate tasks such as architectural pressure testing and automated implementation whilst maintaining a unified audit trail [5]. Industry projections underscore the speed of this adoption, with Gartner predicting that 40% of enterprise applications will include embedded A-AI agents by the end of 2026 [6].

However, the transition from individual agents to fully embedded agentic systems introduces a qualitatively different governance challenge. Whilst agent components are increasingly standardised and widely adopted, systems that are authorised to execute tasks over time within organisational workflows remain comparatively immature. This rapid adoption is not matched by an equivalent maturation of organisational governance. The same Gartner forecast predicts that over 40% of A-AI projects will be cancelled by the end of 2027, citing escalating infrastructure costs, unclear business value, and inadequate risk controls as the primary causes [6]. A recent industry security report reinforces this: whilst most surveyed organisations plan to deploy A-AI into business functions, only 29% report being prepared to secure those deployments [**cisco2026aisecurity**].

This governance gap is further compounded at the strategic level. Across industries, decision-makers often lack the expertise required to meaningfully oversee AI deployments. Surveys of senior leadership indicate that many boards have limited familiarity with AI, and in a significant number of organisations it is not consistently addressed in governance discussions [7]. At the same time, AI is already in use across a large majority of organisations, with more than 88% reporting adoption in at least one business function, creating a disconnect between deployment and oversight [7]. As organisations transition from advisory AI applications to A-AI systems that can act over time within organisational workflows, this disconnect becomes operationally significant.

Existing AI governance frameworks already provide substantial guidance on risk classification, documentation, oversight, and lifecycle management [8, 9, 10]. These frameworks are primarily designed

to evaluate outputs, assess risk, and ensure compliance at the level of individual systems. However, in A-AI settings, the central question shifts from evaluating outputs to governing action: what the system is permitted to do in practice, which decisions it may execute without human intervention, how its scope of action is bounded, and when changes in that scope require review or re-authorisation.

This shift is not fully accommodated by existing frameworks. They are largely organised around identifiable system boundaries, pre-deployment assessment, and relatively stable allocations of responsibility. Whilst these assumptions are appropriate for many current AI applications, they are less adequate where systems can extend their practical role through new tool access, changing usage patterns, and multi-agent interaction [1, 11]. Even within the emerging literature on governing agentic systems [12, 13], delegated authority is not treated as the organising concept. The problem is therefore not the absence of AI governance as such, but the absence of structured governance for delegated authority within existing practice.

This research argues that delegated authority should be treated as a governance object in its own right. Governance must extend beyond what is useful, safe, or legally compliant to making delegated action explicit: defining what the system may do, what remains outside scope, who is accountable, and when operational change triggers renewed governance attention. Without such governance, formally approved deployments risk drifting away from what was originally intended or understood. Therefore, this research focuses on the following central research question:

> *What enterprise governance architecture is needed to specify, monitor, and revise delegated authority in A-AI systems across their lifecycle?*

To answer this question, the research adopts a Design Science Research (DSR) approach [14]. The problem of delegated authority in A-AI is not well treated as a single-variable compliance problem, but is embedded in what Ackoff [15] describes as a "mess": a set of interrelated organisational, technical, and regulatory problems that cannot be addressed in isolation. This makes the design of a governance artefact appropriate. The research therefore combines problem investigation, artefact design, and evaluation in order to develop a governance architecture that renders delegated authority explicit, bounded, monitorable, and revisable over time.

Figure 1.1 positions the Delegation Governance Architecture (DGA) within the broader AI governance landscape. Regulatory instruments define legal obligations, whilst governance frameworks and organisational principles translate these into management commitments. Between these commitments and concrete operational decisions about what a system may do lies an operationalisation gap: high-level requirements are not directly actionable as authority specifications. The DGA is designed to occupy this gap.

This research makes four main contributions. First, it reconceptualises governance of A-AI as a problem of delegated authority rather than treating it solely as an extension of model risk, ethics, or compliance. Second, it identifies and structures five governance gaps that arise when authority is delegated to adaptive systems whose practical role may evolve after deployment. Third, it develops the DGA as a design artefact that specifies authority boundaries, monitoring and enforcement conditions, and lifecycle revision logic. Fourth, it grounds this architecture in enterprise practice through engagement with CGI, demonstrating both the applicability of the artefact and the organisational conditions required for its use.

The remainder of the research is structured as follows. Chapter 2 analyses the governance problem and develops the case for treating delegated authority as a distinct governance object. Chapter 3 explains the research methodology used to investigate the problem, design the artefact, and evaluate it. Chapter 4 presents the DGA. Chapter 5 evaluates the artefact through applied and practitioner validation. Chapter 6 discusses the contribution, limitations, and implications of the research.

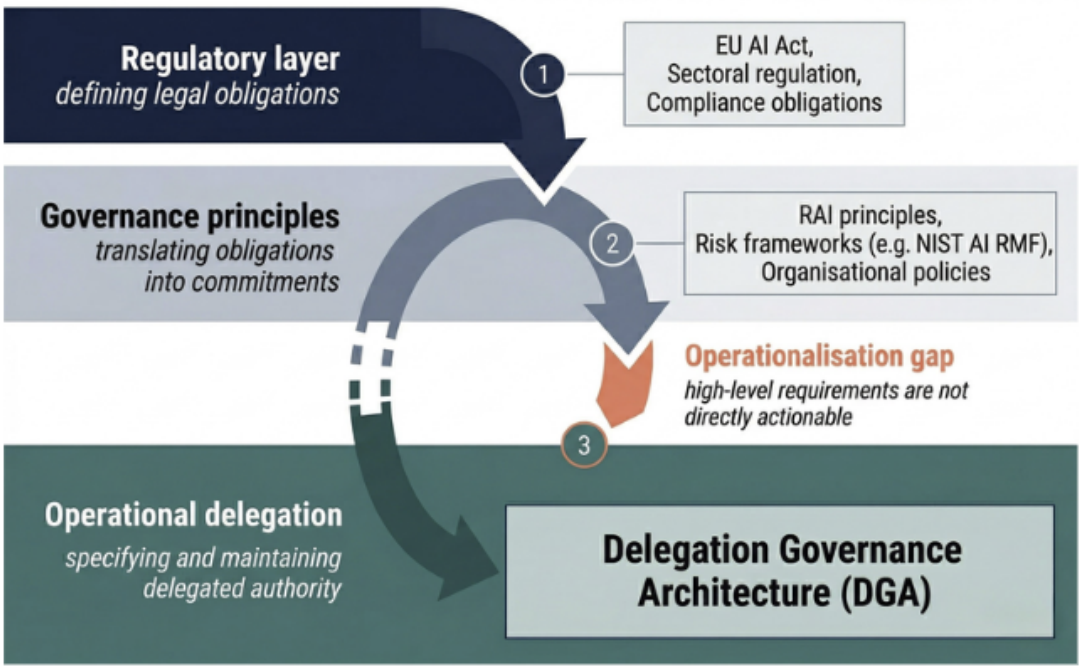

Figure 1.1: Positioning of the Delegation Governance Architecture (DGA) within the (agentic) AI governance stack. Regulatory instruments and governance principles define high-level requirements and commitments. The DGA addresses the translation and operationalisation gap by bridging these commitments with actionable mechanisms for specifying and maintaining delegated authority.

Chapter 2

Governing Delegated Authority in Agentic AI: Problem Context

Artificial intelligence governance has developed rapidly over the last decade. A growing set of normative principles, regulatory initiatives, standards, and enterprise control practices aim to ensure that AI systems are developed and used responsibly. However, the literature is more strongly focused on articulating normative commitments than on explaining how those commitments are operationalised across the design, deployment, monitoring, and evaluation of AI applications. Responsible AI (RAI) research has concentrated heavily on high-level principles, whereas the practices by which those principles are translated into implementable governance remain comparatively underdeveloped and fragmented [16, 17, 18, 19]. With current attention shifting from advisory and content-generation uses towards more agentic forms of AI, this gap becomes particularly acute.

Recent literature characterises A-AI as a shift from prompt-based response generation and single-agent task execution towards systems that can pursue goals over time, adapt behaviour with limited human intervention, and, in more advanced forms, decompose tasks, invoke tools, and coordinate across multiple agents [1, 11, 20, 21, 22]. Practically, this is a shift from systems that primarily *inform* human action to systems that may *participate in* or *execute within* organisational workflows. Whilst current implementations predominantly rely on large language models (LLMs) as the reasoning component within these execution loops, the governance challenge is not specific to any single model architecture. It arises whenever a system is positioned to select and execute actions within organisational processes with reduced human oversight. Consequently, the core governance question changes: it is not only whether an AI output is accurate, safe, or compliant, but also what authority may be delegated to the system, under which conditions, with which constraints, and by which mechanisms that delegation can later be reviewed or revised. In this sense, A-AI introduces a risk-management problem as well as a delegation problem, because organisations increasingly rely on systems that assume a more active role in task execution under conditions of uncertainty [23].

This chapter develops that problem context. It begins by clarifying the analytical vocabulary used throughout the report, distinguishing capability, autonomy, authority, authorisation, and accountability. It then examines how delegated authority is treated in existing AI governance frameworks, explains why A-AI creates a governance problem not fully captured by those frameworks, and illustrates that problem empirically through engagement with CGI's AI governance system. The chapter concludes by synthesising five structural governance gaps that motivate the design of the DGA developed in later chapters.

2.1 Analytical Vocabulary: Capability, Autonomy, Authority, Authorisation, and Accountability

Technical work describes autonomy, goal pursuit, planning, memory, tool use, sequential reasoning, and multi-agent coordination; governance work emphasises oversight, responsibility, transparency, accountability, and the structural, relational, and procedural practices by which organisations operationalise those principles [12, 16, 17, 24]. These strands overlap but are not interchangeable, as they address different aspects of system behaviour and organisational control. For the purposes of this research, it is important to distinguish between what a system *can* do technically, how *autonomously* it may operate, what it is *permitted* to do, how that permission is *authorised*, and who remains *accountable* for the system's actions over time.

Table 2.1 summarises five core concepts used throughout this research. These are introduced to stabilise the analytical vocabulary used throughout the research and to clearly separate technical properties from governance decisions. Given the emerging state of the literature, the definitions are intended as analytic tools rather than universal definitions.

Table 2.1: Core governance concepts used in this research.

Concept	**Definition**
Capability	What the system is technically able to do, given its model, tools, data access, interfaces, and operating environment.
Autonomy	The degree to which the system can select and execute means of action without step-by-step human direction. Autonomy describes the system's ability to act; it does *not* imply permission; it only describes what the system *can* do.
Authority	What the organisation formally permits the system to do within a defined operational context; that is, the scope of delegated authority (or *scope of action*) granted to it. Authority defines what the system is *allowed* to do, the limits and conditions attached to that permission, and the mechanisms for review or revocation.
Authorisation	The governance act through which delegated authority is granted, bounded, conditioned, or withdrawn.
Accountability	The prospective assignment of responsibility to identifiable actors for defining, approving, monitoring, revising, and intervening in the system's exercise of delegated authority.

Capability refers to technical possibility; *autonomy* to the degree of independent operation. These are often treated as defining properties of A-AI [1, 11, 20], but neither resolves the governance question. A system may be highly capable and highly autonomous while operating outside any formally permitted scope, or well within it, depending on how authority is defined. Treating technical properties as if they determine what a system is permitted to do risks replacing a governance judgement with a purely technical description.

Authority is the central concept for this research. It refers to the operational decision space the organisation formally permits the system to act within, often shaped by organisational objectives, risk appetite, legal constraints, and the surrounding control environment. *Accountability* concerns who is responsible for defining, approving, monitoring, revising, and intervening in the system's exercise of delegated authority. It is treated here prospectively, as the formal assignment of responsibility in advance, rather than solely as the attribution of blame after the fact, consistent with the need for structured responsibility and oversight in responsible AI governance [16, 17]. This focus is consistent with literature that

treats interaction with agentic information systems as delegation involving the transfer and coordination of rights and responsibilities between human and non-human actors [23]. These distinctions establish the central governance problem addressed in this research: how to specify, monitor, and revise delegated authority over the lifecycle of A-AI systems.

2.2 Delegated Authority in Existing AI Governance Frameworks

AI governance today is well-developed across regulatory, standards-based, and organisational levels. These instruments provide structured approaches to risk management, documentation, oversight, and post-deployment monitoring. The problem addressed in this research is therefore not the absence of governance, but what that governance is designed to manage. Existing frameworks primarily treat AI systems as objects of risk, compliance, and assurance [8, 9, 10, 16, 17, 18]. They do not systematically treat *delegated authority*, that is, what a system is permitted to do in operational terms, as a distinct governance object [13, 25]. Yet once tasks are delegated to agentic systems, the governance problem increasingly resembles a principal–agent problem, in which limited observability, hidden information, and potential misalignment complicate the principal's ability to control delegated action over time [23, 26].

2.2.1 Governance Instruments and Their Focus

At the regulatory level, the Regulation (EU) 2024/1689 on Artificial Intelligence (EU AI Act) establishes binding requirements for AI systems based on risk classification and intended use. These include obligations for risk management, documentation, logging, human oversight, and post-market monitoring [8]. At the standards level, the AI Risk Management Framework (AI RMF) structures AI risk management across the lifecycle through the functions *govern*, *map*, *measure*, and *manage* [9], whilst ISO/IEC 42001:2023 Artificial Intelligence Management System (ISO/IEC 42001) defines requirements for organisational AI management systems, emphasising controls, accountability, and continuous improvement [10].

Within organisations, these approaches are implemented through policies, review processes, approval gates, monitoring practices, and incident response procedures. Enterprise AI governance can therefore be understood as a combination of structural, relational, and procedural mechanisms that operationalise accountability, transparency, and oversight [16, 17]. In practice, governance is enacted through activities such as risk assessment, system evaluation, staged approval, and ongoing monitoring [27, 28].

Taken together, these instruments provide substantial coverage of how AI systems should be assessed, controlled, and monitored across their lifecycle. However, they are consistently organised around managing system risk and ensuring compliance. Table 2.2 summarises the primary focus of these instruments and how they address post-deployment change.

Across these frameworks, what remains insufficiently specified is the governance of *delegated authority*. Frameworks do not explicitly define or track what a system is *permitted* to do in operational terms, nor how that permission should be revised as the system evolves [13, 16, 17, 25].

This limitation becomes more pronounced in dynamic and agentic contexts. Many frameworks implicitly assume relatively stable system boundaries, within which performance and risk can be assessed. In practice, however, systems may acquire new capabilities, interact with additional tools, and operate across changing environments after deployment [13, 17]. Under these conditions, governance based on static classification, intended purpose, and periodic review becomes insufficient.

As a result, governance remains focused on evaluating systems as artefacts, but they should be viewed as actors operating within an explicitly defined and evolving *scope of action*. The continuous specification and revision of that scope, i.e. delegated authority, is not systematically addressed in existing frameworks.

Table 2.2: Prominent AI governance instruments and their primary governance objects.

Instrument	**Type**	**Primary focus**	**Treatment of post-deployment change**
EU AI Act	Binding regulation	Risk classification, legal obligations, conformity, oversight, and deployer/provider responsibilities	Addresses system change through obligations such as logging, human oversight, quality management, and post-market monitoring, especially for high-risk systems
NIST AI RMF	Voluntary framework	Lifecycle AI risk management through Govern, Map, Measure, and Manage	Treats change as part of iterative risk management and ongoing measurement and management activities
ISO/IEC 42001	Management system standard	Organisational AI governance and control through an AI management system	Treats change through documented controls, management system maintenance, and continual improvement
Enterprise governance models	Organisational practice	Internal review gates, approval processes, monitoring, accountability, and assurance routines	Typically addressed through policy revision, risk reassessment, monitoring, and escalation procedures

2.3 Agentic AI as a Delegated-Execution Shift

The governance gap identified in the previous section arises as AI systems move from advisory roles toward delegated execution within organisational workflows. A-AI should therefore not be understood merely as a more advanced form of GenAI, but as a change in how AI is positioned in socio-technical systems. The critical governance implication is that such systems can select and execute actions over time with reduced human intervention. In this respect, the shift towards A-AI is an organisational redistribution of action, in which systems assume a more active role in executing tasks that would otherwise remain under direct human control [23].

In GenAI deployments the system primarily informs human decision-making; authority remains fully with humans. In A-AI settings, systems may initiate actions, sequence tasks, invoke tools or APIs, and coordinate with other agents inside organisational processes [1, 11, 13, 22]. Figure 2.1 shows this workflow shift from decision-support to workflow execution. Once authorised to act in this way, governance must address more than output quality alone. It must look at what the system is permitted to *do*, which decisions remain human, where escalation is required, and how boundaries are maintained over time.

This shift introduces a temporal dimension to governance. A-AI systems operate through feedback loops that allow behaviour to change over time. As a result, system behaviour is not fixed at deployment, but evolves during operation. Not all changes are governance-relevant. A distinction can be made between *authority-preserving* changes, which occur within an existing authorised scope, and *authority-altering* changes, which modify what the system can effectively do. Authority-altering changes may result from new tool permissions, changes in escalation thresholds, expanded workflow roles, or altered coordination with other agents.

This distinction matters because existing frameworks require logging, monitoring, and review [8, 9, 10], but these mechanisms do not determine whether observed changes remain within the authorised scope or require re-authorisation. The missing element is a mechanism for determining whether a system's behaviour remains within its authorised scope. This requires explicit definitions of authority boundaries, criteria for when deviations trigger escalation, and structured processes for reviewing and re-authorising authority when it changes.

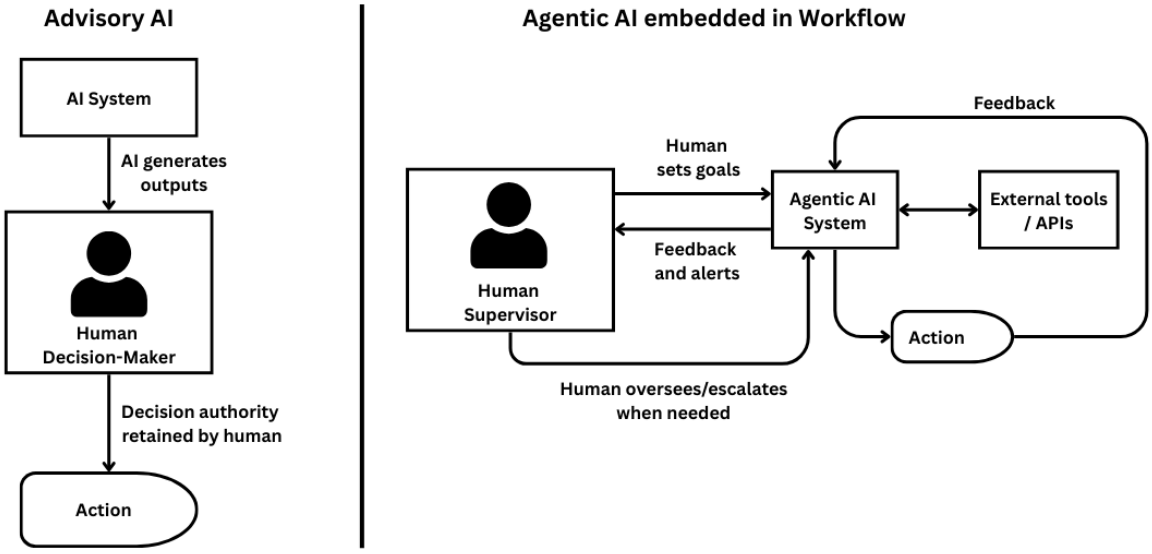

Figure 2.1: Shift from advisory to agentic AI. A shift from output-based systems with human decision authority to workflow-embedded systems that execute actions and adapt through feedback, under human oversight.

2.4 AI Governance Practice at CGI

The governance gap identified above is visible in practice at CGI. CGI is used here as a purposeful empirical setting. The organisation has a relatively mature governance environment, meaning that the limitations observed are more likely to reflect structural issues than simple organisational immaturity (see Chapter 3). CGI has a broad AI governance landscape, including a *RAI framework*, formal risk assessment instruments, lifecycle guidance for AI delivery, an emerging *A-AI Delivery Process*, and internal infrastructure for sharing AI assets and governance materials. Together, these provide substantial coverage of risk management, compliance, and delivery assurance. The question is therefore not whether governance exists, but whether it can be translated into explicit and operational decisions about what authority an A-AI system may exercise, and how that authority is maintained over time.

Governance at CGI is also highly distributed. Responsibility is shared across functions, including the *Global AI Team*, *Engagement Assessment Services (EAS)*, and delivery teams, and operates across *Strategic Business Units (SBUs)* with significant local autonomy. Interview participants consistently described a situation in which governance structures and tools exist, but are unevenly known, accessed, and applied across projects and business units (I2, I8, I9). This distinction is important as it reinforces that the issue is not the absence of governance artefacts, but how they are organised and used in practice.

Three patterns stand out. First, CGI has a substantial governance environment, but awareness and use of that environment are inconsistent. Participants described a real governance structure around AI, including councils, risk tools, and management frameworks, but also emphasised that many practitioners do not know which instruments exist, cannot easily access them, or do not use them in practice (I2, I8, I9). As one participant noted, the framework is regarded as solid, but it is not embedded in a way that makes its use routine (I9). In other words, governance exists, but it is not consistently activated.

Second, the main breakdown occurs in translating governance artefacts into concrete decisions. The *RAI framework* provides high-level principles such as transparency and accountability, and the *AI Risk Matrix* supports structured risk assessment. However, these do not by themselves answer practical questions such as what a system should be allowed to do, when it should escalate, or how its authority should be bounded (I2, I7, I9). Practitioners described these tools as useful starting points, but also as insufficient

Table 2.3: CGI governance practice and delegation-relevant limitations.

Instrument	**Governance role**	**Delegation-relevant limitation**
Responsible AI (RAI) framework	Provides organisation-wide principles, oversight commitments, and regulatory alignment.	Establishes expectations, but does not directly translate them into explicit authority boundaries, escalation thresholds, or re-authorisation rules.
AI Risk Matrix / ARIA	Provides structured risk assessment and mitigation planning during opportunity pursuit and delivery.	Assesses solution risk, but does not formally define what authority is delegated to the system in operation or when that authority should be revisited.
General AI lifecycle / Agentic AI Delivery Process	Provides lifecycle guidance for design, integration, deployment, and operation of AI systems.	Indicates where governance should occur, but does not provide mechanisms for specifying, monitoring, or revising authority.
AI Exchange and internal knowledge-sharing infrastructure	Supports sharing of templates, examples, governance materials, and reusable assets.	Improves visibility and reuse, but remains fragmented and partly dependent on informal requests, limiting consistent adoption and organisational learning.

for resolving use-case-specific questions (I7). As a result, authority decisions are often made through design choices, implementation constraints, and local judgement rather than through explicit governance artefacts. As organisations move towards more agentic forms of AI, these decisions are likely to become embedded in configuration and workflow design unless they are explicitly governed.

Third, governance becomes weaker after deployment. CGI's lifecycle frameworks indicate where governance activities should take place, but they do not provide a clear logic for assessing whether changes during operation remain within an authorised scope or expand it. Participants described governance as heavily focused on early-stage assessment, with less structured attention to what happens once systems are in use. In practice, systems are typically developed and extended incrementally as use cases mature, edge cases emerge, and additional functionality is introduced (I1, I3, I6, I11). However, there are no consistent mechanisms to determine whether these changes remain within authorised boundaries or require renewed governance attention.

This issue is reinforced by existing oversight arrangements. Human-in-the-loop (HITL) is widely used as a safeguard, reflecting an attempt to retain familiar accountability structures. However, participants questioned whether this remains sufficient in more autonomous settings. One participant described the human role as "not really in the loop, but rather standing outside watching the loop" (I1), highlighting the limited basis for meaningful intervention. At the same time, responsibility for design, deployment, monitoring, and change is distributed across multiple actors, making it unclear who owns authority decisions over time or when intervention should occur (I2, I7). The result is a mismatch: authority is effectively delegated to systems, while accountability remains organised around human-centred processes.

The way knowledge is shared within CGI reinforces this pattern. In addition to formal artefacts, both interviews and direct observation show that knowledge sharing is often informal and distributed. Practitioners frequently rely on colleagues, internal communities such as Teams channels, or previous client work to find examples and guidance (I8, I11). The *AI Exchange* is a recent attempt to centralise this, but it currently complements rather than replaces these practices. As a result, governance knowledge is available, but not always easy to find or consistently applied. This makes it difficult to learn systematically from past deployments and to feed those lessons back into governance practice.

Overall, CGI does not lack AI governance. Instead, it reflects a governance-mature but fragmented environment in which multiple artefacts, roles, and structures exist but are not consistently embedded in practice. The core issue is not the presence of governance, but its translation into explicit, lifecycle-oriented

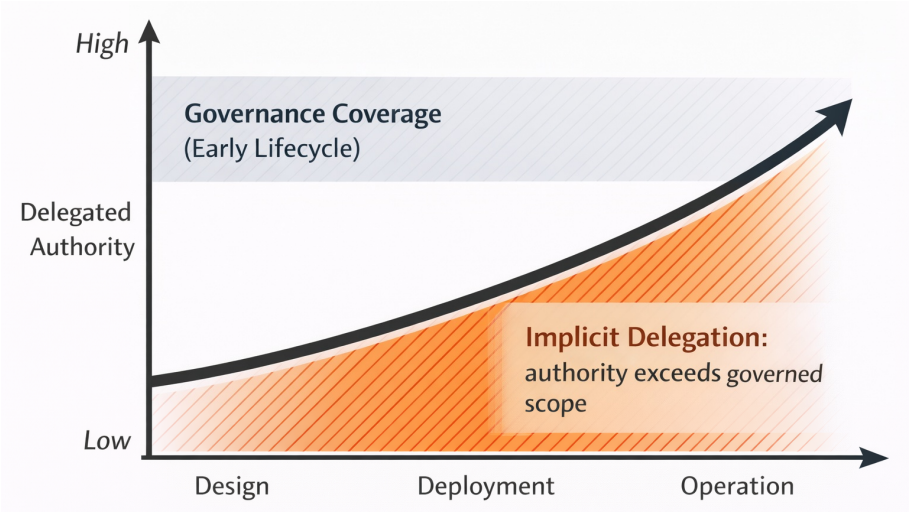

Figure 2.2: Delegated authority versus governance coverage across the AI lifecycle. As systems move from design to operation, delegated authority increases, while governance remains concentrated in early lifecycle stages. The shaded region represents *implicit delegation*: the expanding gap where systems are able to act beyond explicitly specified, monitored, or re-authorised authority boundaries.

decisions about what authority an agentic system may exercise, under which constraints, and how that authority should be revised over time. As a result, delegated authority remains weakly formalised and is often carried implicitly through design choices and operational decisions rather than governed explicitly. This pattern is best described as *implicit delegation*.

2.5 Five Structural Governance Gaps

The analysis so far shows that the problem addressed in this research is not the absence of AI governance, but the absence of governance mechanisms centred on *delegated authority*. Existing frameworks provide substantial coverage of risk management, oversight, documentation, and post-deployment monitoring [13, 16, 17, 25]. CGI likewise has a meaningful governance environment. Yet delegated authority remains weakly specified, unevenly monitored, and insufficiently revised as systems become more agentic and adaptive.

From the literature and the CGI case, five structural governance gaps emerge. They are *structural* because they do not result from a single missing tool or policy, but from how governance is organised across the lifecycle. Table 2.4 summarises these gaps and their main implications.

These gaps are closely connected. If authority is not explicitly defined (G1), it becomes difficult to know when governance should be triggered (G2) or how operational changes should be interpreted (G3). Systems may then evolve beyond their originally intended scope without formal re-authorisation (G4), while accountability remains distributed and difficult to assign clearly in advance (G5).

Table 2.4: Structural governance gaps in delegated authority.

Governance gap	Problem and consequence
G1: Authority boundary specification	Delegated authority is not made explicit, but tends to remain embedded in technical design, workflow configuration, and local implementation choices. As a result, there is no clear baseline for what has actually been authorised, making it difficult to assess whether later changes remain within scope.
G2: Escalation and review	Changes in delegated authority are not consistently linked to governance review. Review may be bypassed, handled informally, or triggered by project stages rather than by changes in the system's effective scope of action, allowing authority-related decisions without structured oversight.
G3: Classification of authority-relevant change	Governance does not clearly distinguish between changes that remain within an authorised scope and changes that materially expand system authority. As a result, organisations may observe system evolution without being able to determine whether it is routine optimisation or a governance-relevant shift.
G4: Re-authorisation over time	No structured process exists to reassess delegated authority once systems are in operation. Incremental changes can therefore accumulate over time without renewed authorisation, causing formal governance decisions to drift away from operational reality.
G5: Accountability allocation	Responsibility for delegated authority is distributed across actors without clear lifecycle ownership. This makes accountability more reactive than prospective, and complicates responsibility for defining, monitoring, and revising system authority over time.

2.6 Problem Statement and Research Goal

The analysis in this chapter leads to the following problem statement:

> Current AI governance does not systematically treat *delegated authority* as an explicit object of governance. Existing frameworks and enterprise processes focus primarily on risk, compliance, and periodic review, but lack the governance logic needed to specify what an agentic AI system is permitted to do in operation, to distinguish between authority-preserving and authority-altering change, and to trigger structured review or re-authorisation when a system's effective scope of action shifts. As a result, delegated authority remains weakly formalised and can expand in practice without corresponding governance decisions.

In response, this research aims to design and evaluate the DGA: a governance artefact that treats delegated authority as a lifecycle object. The purpose of the DGA is to make delegated authority explicit, support structured authorisation decisions, distinguish between operational change within scope and change that expands authority, define when escalation or re-authorisation is required, and allocate accountability prospectively across the lifecycle.

The DGA is considered successful if it addresses the five structural governance gaps identified in this chapter and provides a coherent and implementable basis for specifying, monitoring, and revising delegated authority in adaptive A-AI systems.

Chapter 3

Research Methodology

This research applies DSR as its overarching methodology [14]. The problem identified in Chapter 2 is a design problem: organisations need governance mechanisms for delegated authority in A-AI that are not yet available in current practice. The study was therefore organised into three phases: problem investigation, treatment design, and evaluation. Figure 3.1 provides an overview of this structure.

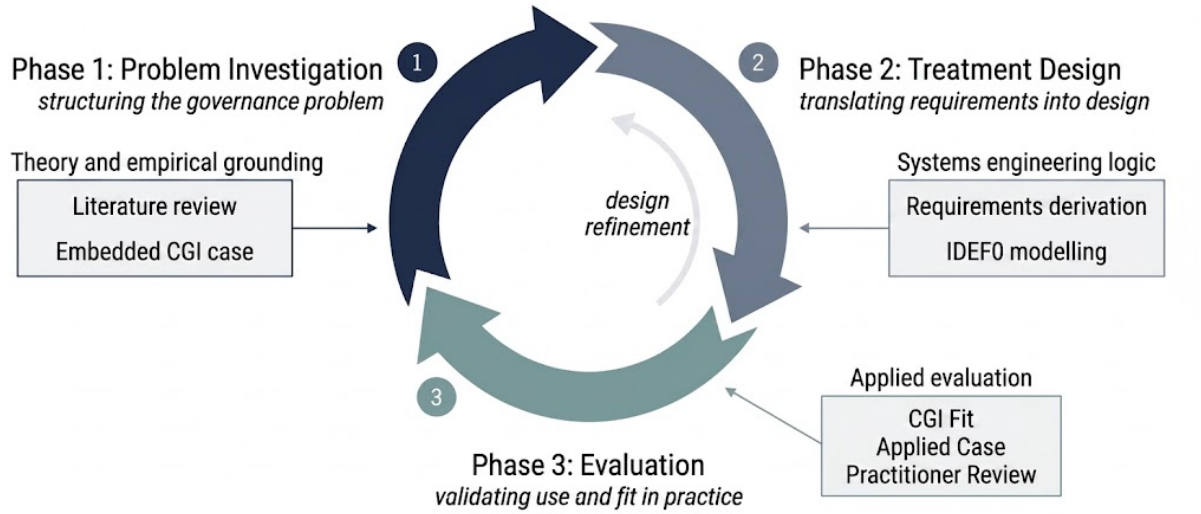

Figure 3.1: Design Science Research (DSR) process applied in this research. The three phases, problem investigation, treatment design, and evaluation, are connected through an iterative feedback loop, where insights from later stages inform and refine earlier ones.

Problem investigation was used to diagnose and structure the governance problem; treatment design translated that diagnosis into the proposed artefact; and evaluation assessed its expected utility. This logic is consistent with Wieringa [14], who positions design research as a movement from understanding the problem context, through the design of an artefact, towards a justified assessment of that utility. The methods used within the study were not confined neatly to a single phase. In particular, the embedded case study at CGI spans the full research design, and both the literature review and the problem-structuring work informed more than one stage of the inquiry. Table 3.1 details the methods used within each phase and their main outputs. The remainder of this chapter explains each methodological component in turn.

3.1 Embedded Case Study Strategy

The empirical component of this research took the form of a single embedded case study at CGI, with the Netherlands business unit as the primary access point. CGI functions as the empirical setting across the

Table 3.1: Summary of the research phases.

Phase	Methods	Main output	Reported in
1. Problem investigation	Focused literature review; exploratory embedded single-case study at CGI Netherlands; systems-informed problem structuring; iterative abductive analysis	Five governance gaps (G1–G5); root definition; conceptual activity model	Chapter 2; Appendices A, B
2. Treatment design	Requirements derivation from G1–G5, supported by the SSM conceptual activity model; systems engineering design; IDEF0 modelling	Design requirements; DGA; DDA; architectural specification	Chapter 4; Appendix C
3. Treatment evaluation	Analytical traceability; applied validation using Lazăr [29]; practitioner validation	Traceability matrix; applied validation findings; practitioner assessment	Chapter 5

full research design: it grounds the diagnosis of the governance problem, provides the enterprise reference context for the artefact design, and supplies the organisational setting for the evaluation. The case study therefore operates as a continuous empirical thread rather than as a method used only in one phase.

Case study research is appropriate when the research question asks *how* or *why*, when the phenomenon cannot be separated from its organisational context, and when the researcher has limited control over events [30, 31]. All three conditions apply here. Although the central research question is formulated as a design-oriented *what* question, answering it requires understanding how governance is enacted in practice and why existing governance arrangements fall short. The phenomenon of interest, namely the governance of delegated authority, is embedded in and shaped by organisational structures, processes, artefacts, and roles. The researcher also had no ability to manipulate governance arrangements experimentally.

The choice for a single case was made for reasons of analytical depth rather than breadth. The study does not seek to estimate how frequently governance problems occur across organisations, but to diagnose and structure the problem in sufficient depth to inform artefact design. CGI provided a strategically relevant case because it offered access to a governance environment that was sufficiently mature and complex to make structural limitations visible. CGI operates in regulated environments, has an established AI governance landscape, and was actively engaging with A-AI during the study period. If governance gaps around delegated authority are visible in such a setting, they are more likely to reflect structural characteristics of current governance approaches than artefacts of simple organisational immaturity.

The case study was both exploratory and embedded [30]. It was exploratory because governance practice for A-AI remains emergent, and embedded because multiple elements of the governance system were examined within a single organisational context, including governance artefacts, lifecycle processes, organisational roles, and practitioner perspectives. The unit of analysis was CGI's organisational AI governance system: the set of structures, processes, roles, and informal norms through which AI systems are currently assessed, authorised, monitored, and managed. This unit was defined broadly because the governance of delegated authority does not yet exist as a formally bounded organisational practice; what could be studied empirically was the broader governance system and the structural absence within it.

Overall, CGI functions as a strategically relevant case through which structural limitations in current governance approaches become visible. The governance gaps derived from it are thus treated as analytically grounded propositions about how delegated authority in A-AI is under-specified, weakly operationalised, and insufficiently revised in enterprise practice. These propositions are intended as transferable conceptual insights for similar organisational contexts, whilst remaining open to further testing through comparative case research. Figure 3.2 visualises how the embedded case study operates as a continuous empirical setting that supports problem structuring, analysis, design, and evaluation throughout the research.

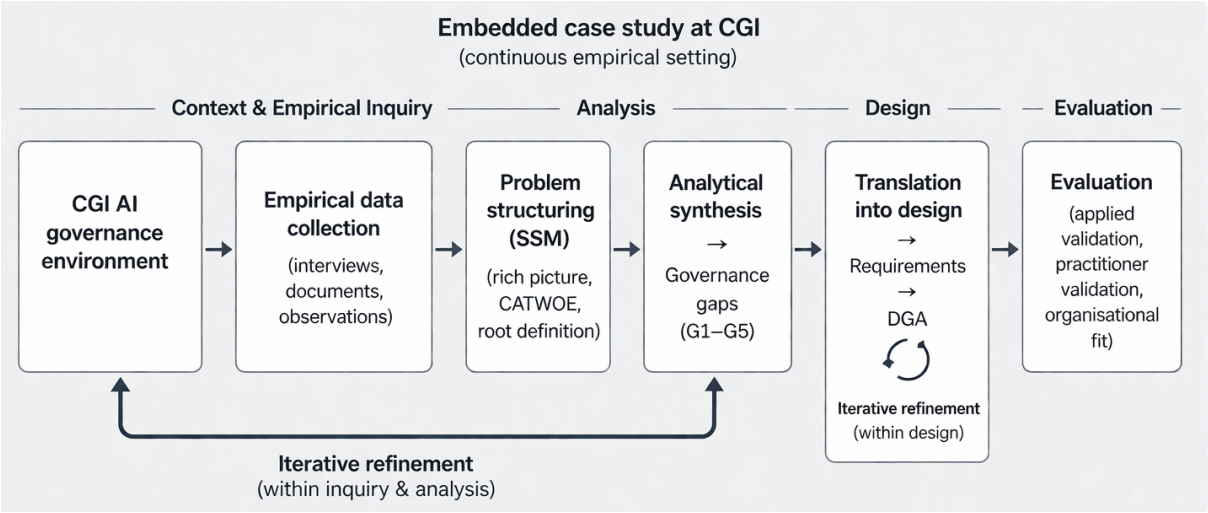

Figure 3.2: Embedded case study process and its role in the research. The case study at CGI provides a continuous empirical setting that underpins problem structuring, analysis, and evaluation. Empirical data collection and problem structuring proceed in parallel and are iteratively refined, leading to the identification of governance gaps. These are translated into design requirements and the artefact, followed by evaluation within the organisational context.

3.2 Focused Literature Review

The literature review focused on AI governance, RAI, standards, accountability, and emerging work on agentic systems. It was focused and iterative rather than formally systematic, which was appropriate given that the field is broad, interdisciplinary, and still developing rapidly. As discussed in Chapter 2, the review established the analytical vocabulary used throughout the thesis and provided the initial basis for distinguishing a broad concern with A-AI governance from the more specific problem of governing delegated authority in systems that may act within organisational workflows.

The review served three methodological functions. First, it provided the conceptual basis for the analytical distinctions used throughout the thesis, especially the separation between capability, autonomy, authority, authorisation, and accountability. Second, it supported the initial diagnosis that current governance approaches primarily treat AI as an object of risk and compliance rather than as an object of delegated authority. Third, it informed both the SSM-based problem structuring and the empirical inquiry by supplying the sensitising concepts used in the interview and analysis process. The review remained iterative throughout problem investigation, as empirical findings prompted targeted follow-up reading where concepts appeared under-specified or analytically important.

3.3 Problem Structuring and Empirical Analysis

Problem investigation combined two interrelated inquiry streams. SSM was used to structure the initially broad and messy problem situation into a more precise formulation. Semi-structured interviews and supporting empirical material then tested and refined that formulation through engagement with governance practice at CGI. The two streams proceeded iteratively: the SSM diagnosis informed the interview design, and the interview findings refined the diagnosis.

3.3.1 SSM-Informed Problem Structuring

SSM [32] was used as a problem-structuring approach within the broader DSR process. Early investigation revealed a broad and messy situation involving regulation, organisational governance, technical implemen-

tation, and uncertainty about how much operational authority could appropriately be delegated to A-AI systems. SSM was therefore appropriate because it supports movement from an initially unstructured situation towards a more explicit account of the relevant actors, tensions, and system purpose. Selected artefacts, including a rich-picture style overview, a CATWOE analysis, and a root definition [33], were used to structure the problem situation and inform the subsequent empirical inquiry. These artefacts are presented in Appendix B.

The SSM work contributed to the study in three concrete ways. First, it informed the interview protocol. The interview topics documented in Appendix A were derived in part from the provisional SSM diagnosis: current governance practice, the translation of principles into operational constraints, authority and accountability, system evolution, and oversight. Second, it informed follow-up literature review. As the interviews revealed recurrent tensions around authority boundaries, lifecycle review, and accountability over time, the literature review was extended beyond general RAI governance towards work on agentic systems, delegation, principal–agent relations, and post-deployment change. Third, and most importantly, it narrowed the scope of the study. Early in the research, the problem could have been framed broadly as enterprise governance for A-AI. The SSM analysis showed that such a scope would be too diffuse. What required specific analytical and design attention was the governance of delegated authority: how authority is specified, translated into operational constraints, monitored in practice, and revised as systems evolve. This scoping decision shaped the problem chapter, the governance gaps, and the eventual design of the DGA.

3.3.2 Empirical Data Collection and Analysis

The empirical material combined semi-structured interviews, internal governance documents, knowledge-sharing sessions, and direct observation of selected governance tools. In total, thirteen practitioners participated across eleven interview sessions, selected purposively on the basis of their involvement in governance, delivery, risk, architecture, strategy, or advisory roles related to AI. The interview protocol, participant overview, coding structure, and representative evidence are provided in Appendix A.

The material was analysed using a theoretically informed thematic analysis approach [34] within an overall abductive logic. Initial sensitising concepts were drawn from the literature review and the provisional SSM diagnosis. The interview material was coded deductively using six themes (T1–T6), derived from the developing problem diagnosis. Although these themes were defined prior to coding, they were refined through engagement with the case material. A theme was marked as present only where the material contained sufficiently direct evidence for the underlying concept.

Because the themes were derived from the developing problem diagnosis, they partly pre-structured the governance gaps that the analysis ultimately produced. The relationship between themes and gaps is therefore not purely inductive. Rather, the literature and SSM generated provisional concepts, the interviews tested and refined those concepts through empirical engagement, and the resulting synthesis produced the governance gaps. The themes should therefore be understood as analytical lenses through which the empirical material was examined, rather than as findings that emerged from the data without prior conceptual direction.

Within this logic, the analysis proceeded iteratively. Concepts from the literature guided interpretation of the case material; empirical findings refined those concepts; and targeted follow-up reading was undertaken where the case revealed analytically important but under-specified issues. This abductive movement produced the five structural governance gaps (G1–G5) synthesised in Chapter 2.

3.4 Artefact Design through Systems Engineering

The artefact design drew on systems engineering principles to move from problem diagnosis to requirements and from requirements to functional specification [35]. Systems engineering was appropriate because the problem addressed in this research is not a single control deficiency, but a coordination problem spanning multiple governance functions, decision points, information flows, and lifecycle stages. As Buede [35] argues, systems engineering is particularly suited to situations in which complex systems must first be defined in terms of purpose, requirements, interfaces, and functional relationships before implementation choices are made.

The design process followed a requirement-led approach. First, the five governance gaps were translated into a structured set of design requirements expressing what a governance architecture for delegated authority must be able to do. These requirements are reported in Appendix C.1. Second, the requirements were translated into an architectural specification through functional decomposition using IDEF0. The notation makes explicit the inputs, outputs, and controls associated with each function, and thereby supports clear representation of governance relationships, dependencies, and decision points. Mechanisms were not specified in implementation detail, since the concrete organisational or technical realisation of functions is context-dependent and falls outside the scope of the present research. The main diagrams are presented in Chapter 4, with the remaining diagrams provided in Appendix C.

The design remained iterative throughout. Requirements informed the first functional specification, and the resulting models were then used to refine the requirements further. This iteration helped clarify where governance functions needed to be separated, connected, or made more explicit in order to address the diagnosed problem coherently.

3.5 Validation Methods

The evaluation is primarily ex ante. Because enterprise-scale governance of A-AI is still emerging, the aim was not to measure long-term organisational outcomes, but to assess whether the proposed architecture is coherent, applicable, and perceived as useful in a realistic enterprise setting. Three complementary methods were used, summarised in Table 3.2.

Table 3.2: Validation methods and their contribution to the evaluation.

Method	**Purpose**	**Data / Setting**	**What it assesses**
Applied validation	Examine whether the DGA produces coherent governance outputs when applied to a realistic deployment	CGI-based A-AI deployment described by Lazăr [29]	Practical applicability as a governance specification device
Practitioner validation	Examine whether the governance problem is recognised and the architecture regarded as relevant	Pre-questionnaire (n=6) and structured discussion session (n=3) with CGI DCS practitioners	Perceived relevance, recognisability, and organisational resonance
Organisational-fit analysis	Examine whether the surrounding governance functions assumed by the DGA exist in the CGI context	Mapping of DGA functions onto existing CGI roles, structures, and practices	Organisational plausibility and governance readiness

Together, these methods assessed the DGA from complementary angles: applicability to a realistic deployment proposal, relevance among practitioners, and plausibility within an enterprise governance context. The findings are reported in Chapter 5 and discussed further in Chapter 6.

3.6 Methodological Scope and Limitations

The methodological choices made in this study were appropriate to its design-oriented purpose, but they delimit the claims that can be made. The study relies on a single embedded case, meaning that the governance gaps are treated as analytically grounded propositions rather than empirically established regularities across organisations. Empirical access was also partial: access to internal documentation was uneven, and the SSM artefacts were constructed by the researcher rather than co-produced with stakeholders.

In addition, the phenomenon studied is emergent. Most examples discussed concerned progressively extended systems rather than mature high-autonomy deployments, meaning that the diagnosis rests partly on anticipatory judgement. Finally, the evaluation is ex ante and does not demonstrate sustained operational effectiveness. The implications of these constraints are discussed in Chapter 6.

Chapter 4

Delegation Governance Architecture

Chapter 2 identified five structural governance gaps in how delegated authority is handled in A-AI. Chapter 3 then explained how those gaps were translated into a design artefact. This chapter presents that artefact: the DGA.

The DGA is a general enterprise governance architecture for specifying and maintaining delegated authority in A-AI deployments. It does not govern AI in the broadest sense, nor does it itself implement technical controls, runtime monitoring, or organisational approval processes. Its role is to structure governance decisions by making delegated authority explicit, specifying how that authority is to be monitored and revised, and maintaining a versioned governance record of those decisions.

The chapter is organised around the functional architecture of the DGA as a designed system. IDEF0 diagrams are used throughout to clarify the functional logic of each architectural part. Appendix C contains the full set of interaction diagrams.

4.1 Purpose and Scope of the DGA

The DGA governs *delegated authority*: the scope of action that an A-AI system is permitted to exercise within a defined operational context. This is narrower than general AI governance and more specific than enterprise risk management. The architecture is concerned with what the system is allowed to do, under what conditions that permission holds, how that permission is governed in operation, and how it is revised when operational reality changes.

This object of governance must be distinguished from both technical capability and autonomy. As discussed in Chapter 2, a system may be capable of interacting with tools or executing complex actions without being authorised to do so in a specific deployment. Capability concerns what the system *can* do; authority concerns what the organisation permits it to do. The DGA makes this distinction explicit and governable.

A second defining feature is its temporal character. Delegated authority is not only an ex ante approval decision. It may evolve over time as tool access changes, usage patterns shift, workflows expand, or system capabilities are updated. The DGA therefore treats delegated authority as a lifecycle object rather than as a one-off approval.

In practical terms, the architecture is organised around four governance questions. Table 4.1 summarises them. These questions define the scope of the DGA and organise the governance artefacts through which delegated authority is specified, enacted, and maintained.

These four questions are addressed through four core artefacts, summarised in Table 4.2. The artefacts are not independent; they form a structured governance chain. The *Authority Boundary Specification* defines the outer limits within which delegation may occur. Within those limits, the *Delegated Authority Specification* determines what is actually authorised. The *Monitoring and Enforcement Configuration*

Table 4.1: Governance questions addressed by the DGA.

Governance question	**What the DGA determines or records**
What is the outer limit of permissible action?	The authority boundary within which delegation may occur.
What is the system actually authorised to do?	The delegated authority specification, including permitted actions, escalation-required actions, and delegation conditions.
How is that delegation governed in operation?	The monitoring and enforcement configuration used to observe, escalate, and contain delegated authority in practice.
How is the delegation reviewed and maintained over time?	Lifecycle review decisions and the versioned governance record maintained in the DDA.

then governs how that authorised scope is observed and controlled in operation. Finally, the *DDA* records these decisions, their justification, and their evolution over time.

Table 4.2: Core artefacts of the Delegation Governance Architecture.

Artefact	**Purpose**
Authority Boundary Specification	Defines the bounded action space by translating deployment context into delegation scope, specifying constraints and prohibited action classes, and defining change-significance criteria that trigger governance review.
Delegated Authority Specification	Specifies what is authorised within the boundary by determining a viable delegation space, mapping system capabilities, classifying actions, and defining escalation routing and delegation conditions.
Monitoring and Enforcement Configuration	Defines how delegated authority is monitored and governed in operation, including governance-relevant signals, enforcement logic, and feasibility calibration.
Delegation Decision Artefact (DDA)	Consolidates the artefacts into a versioned governance record capturing decisions, conditions, and lifecycle evolution.

This design addresses two common failures identified in Chapter 2. First, it prevents delegated authority from remaining implicit in system design or process configuration. Second, it ensures that changes in delegated authority are handled explicitly rather than informally or through project memory alone. The artefacts therefore serve two complementary purposes: they make authority governable in practice and governance decisions traceable over time.

4.2 System Boundary and Context

The DGA is modelled as the system-of-interest within a broader organisational governance context. That context is structured across three layers, each representing a distinct level of governance responsibility. At the highest level, enterprise governance functions define strategic intent, governance policy, and portfolio oversight for A-AI (Appendix Figure C.1). At the middle level, deployment lifecycle functions manage the proposal, design, build, and operation of individual A-AI deployments (Appendix Figure C.2). At the lowest level, deployment governance functions translate deployment proposals into governance specifications, monitor runtime operations for governance-relevant events, and validate and authorise deployments. The DGA sits within this innermost layer (Figure C.3).

This layered framing clarifies the analytical status of the DGA. The architecture is not intended to replace enterprise governance or deployment management. Rather, it occupies a more specific governance role within them: it provides the internal governance logic through which delegated authority is specified, monitored, and revised at deployment level.

4.2.1 Lowest-level deployment governance context

Figure C.3 presents the lowest-level interaction diagram surrounding the DGA. It defines the system boundary by showing what enters the DGA (function 0), what constrains it, and what it produces.

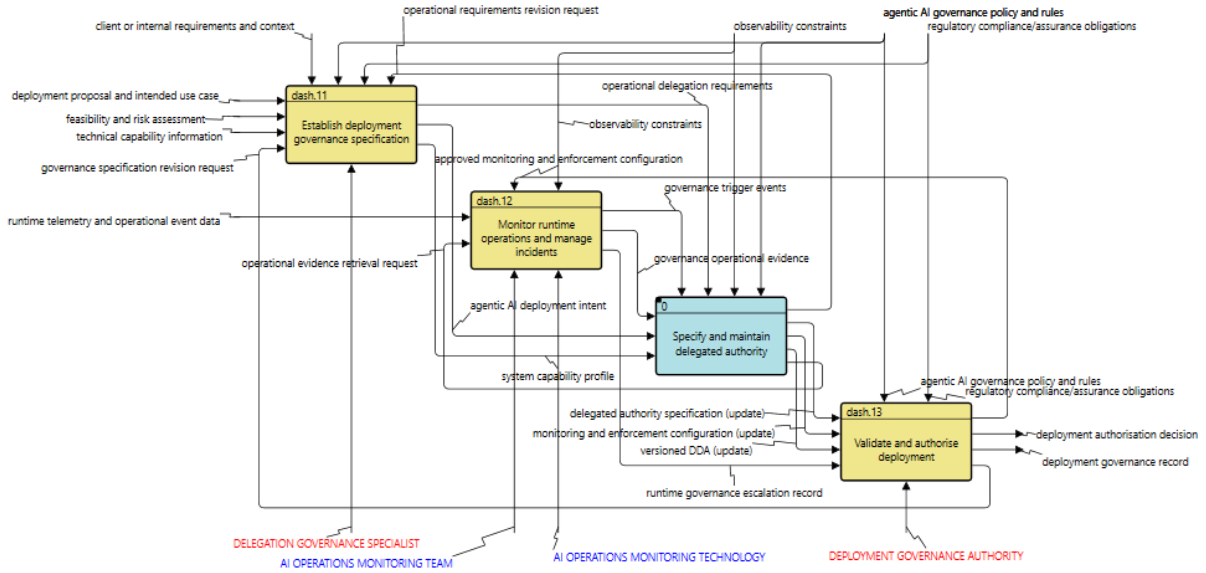

Figure 4.1: Deployment governance functions surrounding the DGA (lowest-level context). The system-of-interest is shown in blue; surrounding governance functions in yellow. Labels along the bottom identify the organisational roles or technology responsible for each function. Inputs enter from the left, controls from above, and outputs exit to the right.

The three surrounding functions each play a distinct role. Function dash.11 (*Establish deployment governance specification*) translates deployment proposals, feasibility assessments, and technical capability information into the operational delegation requirements and observability constraints within which the DGA must work. Function dash.12 (*Monitor runtime operations and manage incidents*) ingests runtime telemetry, applies the approved monitoring and enforcement configuration, and produces the governance trigger events and governance operational evidence that activate and inform the DGA across the lifecycle. Function dash.13 (*Validate and authorise deployment*) receives the governance artefacts produced by the DGA and makes the formal authorisation decision, producing a deployment authorisation decision and a deployment governance record.

These surrounding functions delimit the role of the DGA. The architecture does not itself grant or withhold approval, nor does it operate the runtime controls through which enforcement is enacted. Its role is to structure the governance basis on which those surrounding functions depend.

4.2.2 Boundary Decisions and Modelling Logic

Several boundary decisions warrant explanation. Governance trigger events are modelled as controls rather than inputs because they initiate the governance response without being transformed by it. The response itself is constrained by enterprise governance policy and regulatory obligations, which shape what kinds of delegation are permissible and how governance review must be conducted.

Observability constraints also enter as a control. This reflects the fact that the DGA cannot specify monitoring requirements independently of the technical infrastructure available to support them. In other words, the architecture cannot require observation of behaviours that the deployment context cannot feasibly capture.

The operational evidence retrieval request is modelled as an output directed back to the operational context. This indicates that the DGA is not limited to passively receiving whatever telemetry happens to be available. It can request specific evidence for governance purposes where further clarification or review is needed.

The DGA is therefore positioned as a governance-structuring system: it translates contextual constraints and operational signals into explicit, reviewable delegation decisions while remaining distinct from both runtime execution and formal authorisation.

4.3 Delegation Governance Architecture

Having established the purpose, scope, and system boundary of the DGA, this section turns to the architecture itself. The DGA is intended to be a structured governance system whose functions are enacted by organisational roles and tools within the enterprise. Its contribution lies in the logic through which delegated authority is bounded, specified, governed in operation, and revised over time.

4.3.1 Architecture Overview and Lifecycle Logic

At first-level decomposition, the DGA consists of four interacting functions. Three of these are specification functions: they define the authority boundary within which delegation may occur, determine what authority may actually be delegated within that boundary, and specify how that delegated authority is to be monitored and governed in operation. The fourth function, authority lifecycle calibration, operates across those specification activities. It ingests governance trigger events and operational evidence, assesses whether the current governance state remains adequate, and issues review decisions or targeted revision requests where change is required. Figure C.4 presents this first-level functional decomposition.

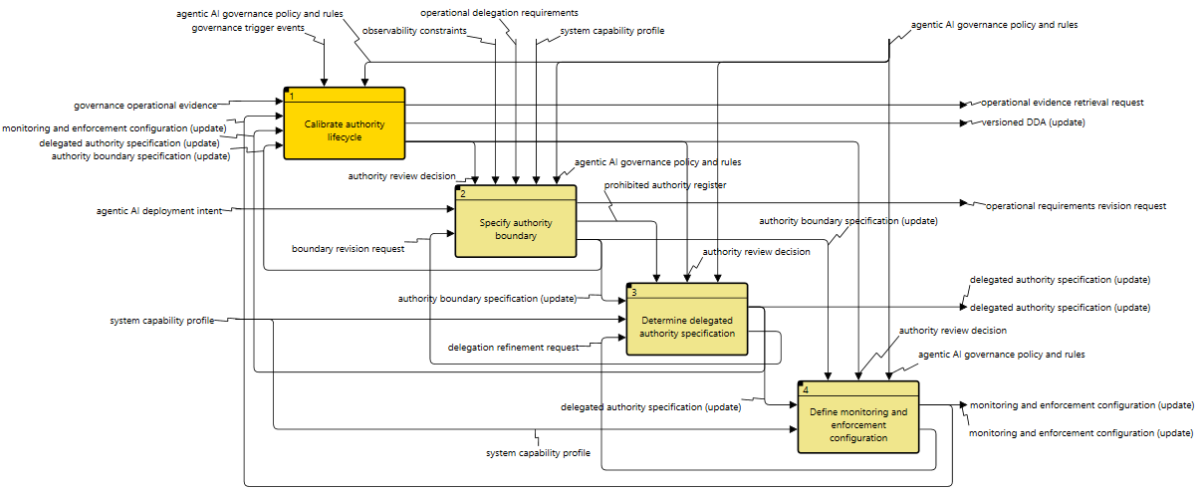

Figure 4.2: First-level functional decomposition of the DGA. The lifecycle calibration function (F1, darker shading) operates across the three specification functions (F2–F4, lighter shading), ingesting governance trigger events and operational evidence, determining whether review is required, and issuing authority review decisions or targeted revision requests. Together, the functions update the versioned DDA and may request additional operational evidence where the existing evidence base is insufficient.

The decomposition shows that the DGA is not a one-off approval flow, but a lifecycle governance system. The specification process begins with the authority boundary function, which translates deployment intent, operational delegation requirements, policy constraints, and capability-related considerations into the bounded action space within which delegation may be considered. Within that boundary, the delegated

authority function determines what may actually be authorised, using the system capability profile and any applicable review constraints to define a viable delegated scope. The monitoring and enforcement function then specifies how that scope is to be governed in operation, producing the configuration through which escalation conditions, prohibited actions, and governance-relevant signals can be observed and acted upon.

Authority lifecycle calibration sits across this chain. It receives governance operational evidence together with updates to the authority boundary, delegated authority specification, and monitoring and enforcement configuration. On that basis, it determines whether the current governance state remains acceptable or whether revision is required. Where relevant change is identified, it can issue an authority review decision, trigger a boundary revision request, initiate delegation refinement, or request additional operational evidence from the surrounding governance environment. In this way, the architecture supports both initial specification and subsequent recalibration without collapsing governance into continuous re-authorisation.

This functional logic enables different governance outcomes. A deployment may be supported within the proposed scope, supported conditionally under narrower or more closely monitored conditions, or deferred pending stronger evidence, improved observability, or revised operational requirements. The DGA itself does not perform the act of authorisation. Rather, it structures the governance basis on which authorisation rests. Formal validation and authorisation are carried out by the surrounding governance function (dash.13, Figure C.3), using the artefacts produced by the DGA as the decision basis.

Across the lifecycle, the architecture produces two outward-facing governance specifications and one persistent governance record. The *Delegated Authority Specification* defines the authorised scope as determined through governance. The *Monitoring and Enforcement Configuration* defines how that scope is to be governed in operation. The DDA records, justifies, and versions these outputs together with the review decisions through which they change over time. Where the available evidence is insufficient to support review or revision, the DGA may also issue an operational evidence retrieval request before a broader authority decision is made.

4.3.2 Authority Boundary Specification

An important function of the architecture is to specify the authority boundary: the structured limits within which delegation may be considered. It produces the Authority Boundary Specification. Figure C.7 presents the functional decomposition.

The process begins by translating the agentic AI deployment intent into a delegation scope: a structured representation of the action space within which delegation could in principle occur. That scope is then constrained through authority boundary constraints, which determine the conditions under which delegation can be considered governable. This step uses the delegation scope together with policy constraints, observability constraints, and the system capability profile. Where the proposed boundary cannot be sustained within the available infrastructure, the function signals this explicitly.

In parallel, a prohibited authority register is established. This identifies classes of action that must not be delegated regardless of technical capability, drawing on governance policy and regulatory prohibition requirements. Where a proposed prohibition cannot be reliably enforced, this is captured as feasibility feedback for use in boundary refinement.

Finally, change-significance criteria are defined. These specify when observed changes in behaviour, usage, or operational context must be treated as governance-relevant and trigger review. The criteria are derived from the delegation scope, the boundary constraints, and the prohibited authority register, and are shaped by regulatory requirements and governance trigger rules. Where the operational delegation requirements provided to the function prove insufficient, an operational requirements revision request is issued to the surrounding governance environment (dash.11, Figure C.3).

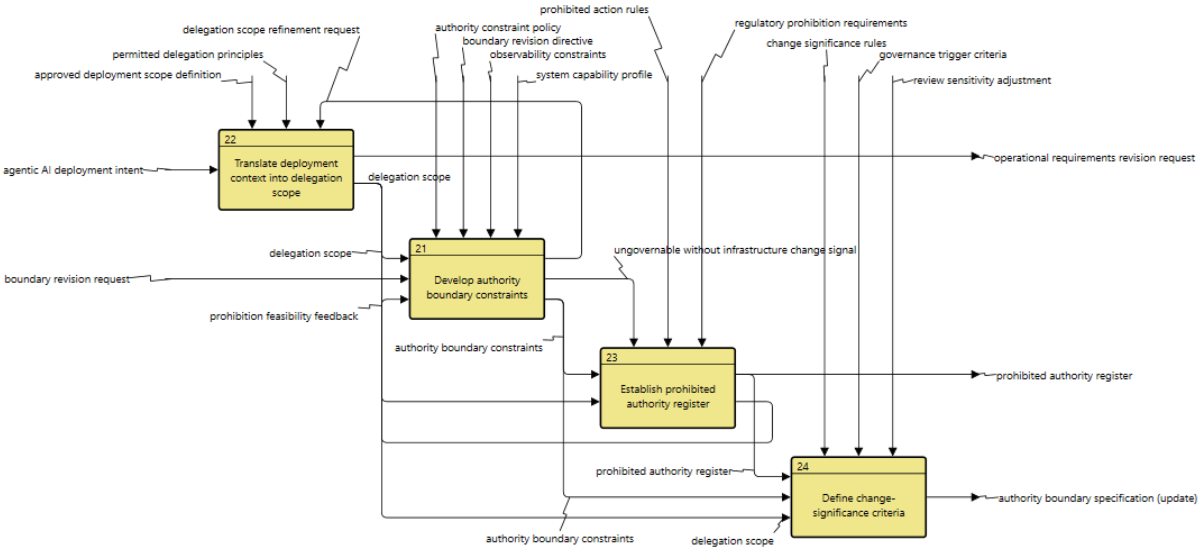

Figure 4.3: Functional decomposition of "Specify authority boundary" (F2). Four sub-functions translate deployment intent into a delegation scope, develop boundary constraints, establish the prohibited authority register, and define change-significance criteria.

The boundary therefore does more than restrict the scope of delegation. It also defines when that scope must be reconsidered. The combined output, the Authority Boundary Specification (update), consolidates the delegation scope, boundary constraints, prohibited action classes, and change-significance conditions into a single governance artefact. This specification then serves as the outer constraint within which delegated authority may subsequently be determined.

4.3.3 Delegated Authority Specification

Within the authority boundary, the DGA determines what is actually authorised. This produces the Delegated Authority Specification. Figure C.8 presents the functional decomposition.

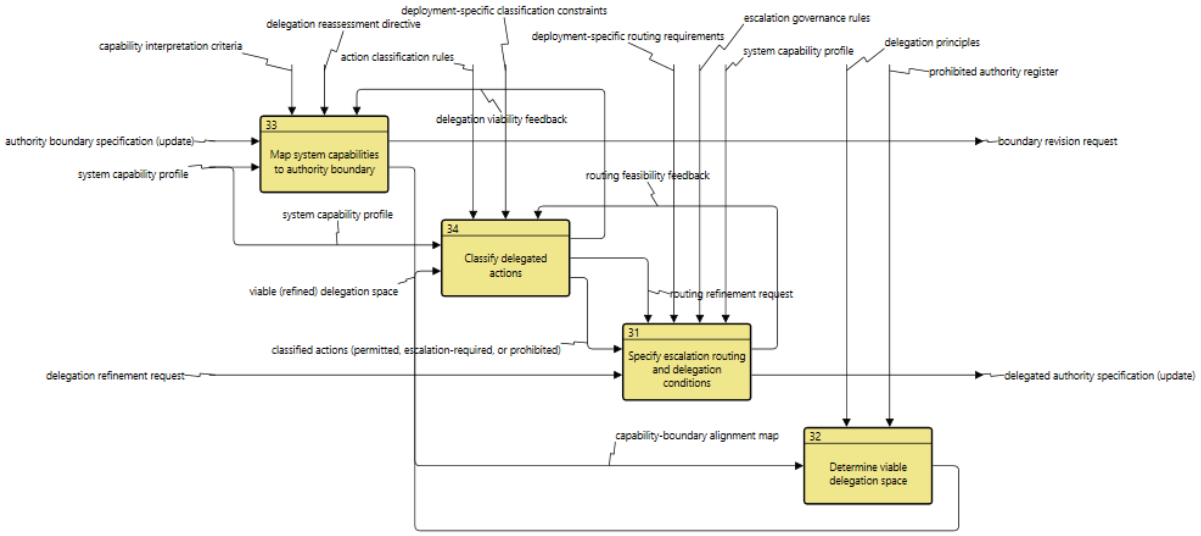

Figure 4.4: Functional decomposition of "Determine delegated authority specification" (F3). Four sub-functions map system capabilities to the authority boundary, determine the viable delegation space, classify delegated actions, and specify escalation routing and delegation conditions.

The process begins by mapping the system capability profile against the Authority Boundary Specifi-

cation. This identifies where capabilities fall within the boundary, where they exceed what may currently be delegated, and where the boundary itself may require reconsideration. Where substantial misalignment is identified, the function may issue a boundary revision request.

On that basis, the architecture determines the viable delegation space: the subset of the bounded action space that can credibly be considered for delegation once capability, prohibition, and governance constraints are taken together. Actions within that space are then classified as permitted, escalation-required, or prohibited, translating a relatively abstract delegation space into explicit governance categories. Finally, escalation routing and delegation conditions are specified, determining under what conditions actions may proceed autonomously, when escalation is required, and how it should be routed within the deployment context.

These functions are logically sequential but not strictly one-way. Classification may reveal that the viable delegation space requires refinement, and routing specification may show that nominally valid classifications are not operationally workable without adjustment. The resulting Delegated Authority Specification (update) therefore records not only what is authorised, but also the conditions and escalation logic attached to that authorisation. It provides the immediate basis for defining how delegated authority will subsequently be monitored and enforced in operation.

4.3.4 Monitoring and Enforcement Configuration

Once delegated authority has been specified, the architecture defines how that authority is to be governed in operation. This produces the Monitoring and Enforcement Configuration. Figure C.9 presents the functional decomposition.

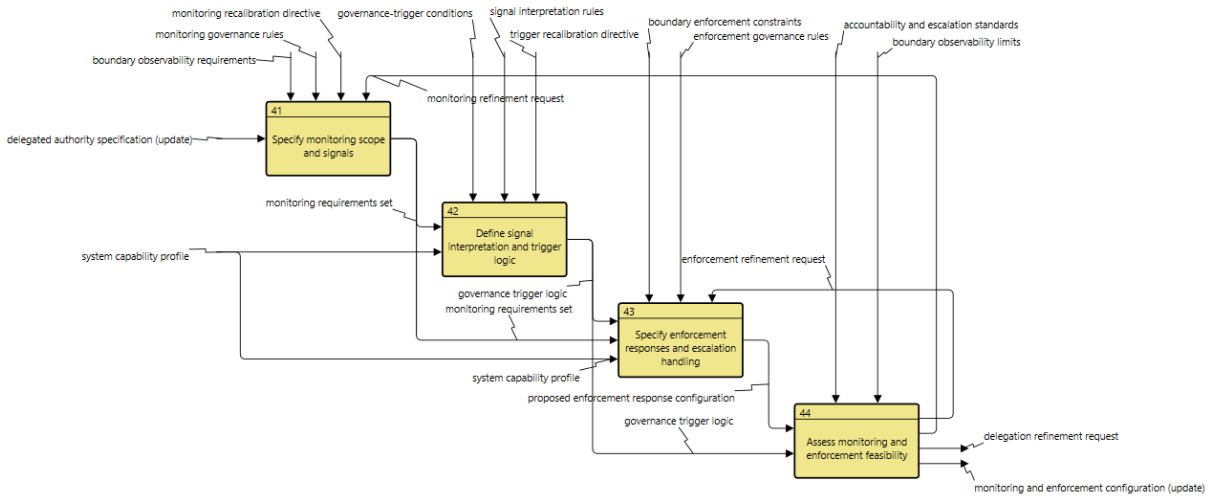

Figure 4.5: Functional decomposition of "Define monitoring and enforcement configuration" (F4). Four sub-functions specify monitoring scope and signals, define signal interpretation and trigger logic, specify enforcement responses and escalation handling, and assess the feasibility of the resulting monitoring and enforcement configuration.

The process begins by determining which actions, states, and interactions must be observable in order to govern delegated authority in practice, producing a monitoring requirements set derived from the Delegated Authority Specification and boundary observability requirements. Those requirements are then translated into signal interpretation and trigger logic, which specifies how observed signals are to be interpreted and under what conditions they should activate escalation, containment, or review. On that basis, the architecture specifies enforcement responses: how escalation-required actions are to be routed, how prohibited actions are to be contained, and how governance responses are to be handled in the deployment context.

The final step assesses whether the proposed monitoring scope, trigger logic, and enforcement responses can be realised consistently within the available operational environment. Where the arrangement proves infeasible, the function may issue a delegation refinement request, indicating that the delegated authority itself requires revision if it is to remain governable.

Monitoring and enforcement are treated here as conditions of delegated authority rather than as a separate governance layer. Delegation is only considered governable where the authorised scope can be meaningfully observed, interpreted through explicit trigger logic, linked to defined governance responses, and shown to be operationally feasible. Where these conditions are not met, the architecture constrains what may be delegated rather than proceeding with an unmonitorable scope.

4.3.5 Lifecycle Calibration, Review, and Revision

The most important function of the DGA ensures that delegated authority remains governed across the full lifecycle. After deployment, the system operates under its current governance specifications until governance trigger events, operational evidence, or policy-based review requirements indicate that the current governance state may no longer be adequate. The DGA therefore does not continuously re-authorise the system; it is activated when relevant change signals are received. Figure C.5 presents the functional decomposition.

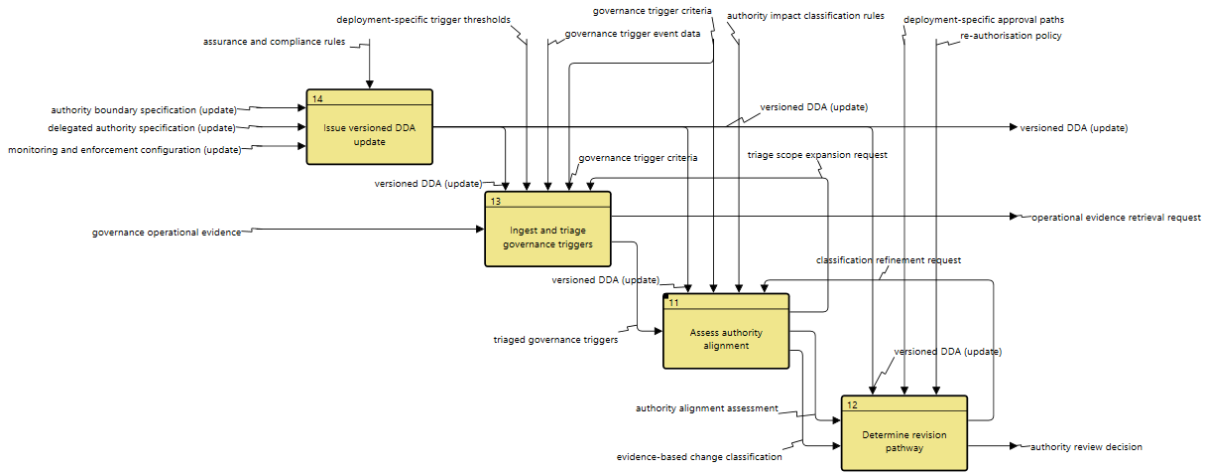

Figure 4.6: Functional decomposition of "Calibrate authority lifecycle" (F1). Four sub-functions ingest and triage governance triggers, assess authority alignment, determine an authority review decision, and issue a versioned update of the DDA.

The process begins with the ingestion and triage of governance triggers, assessing incoming evidence against deployment-specific trigger thresholds and the current governance baseline. Where the available evidence is insufficient, the architecture may issue an operational evidence retrieval request before further review proceeds. Where review is required, the DGA assesses authority alignment. This is the most analytically developed step in the lifecycle function and is decomposed further in Figure C.6.

The assessment proceeds in three steps. First, incoming evidence is mapped to the current governance state, producing a governance state deviation map. Second, the significance of identified deviations is assessed, distinguishing expected operational variation from governance-relevant change. Third, the authority impact of the observed change is classified, determining whether the system remains aligned with its current delegated authority, whether alignment is uncertain, or whether material misalignment has occurred. These steps include targeted feedback loops: where classification reveals insufficient grounding, significance may be reassessed, and where the evidence base proves inadequate, the triage scope may be expanded.

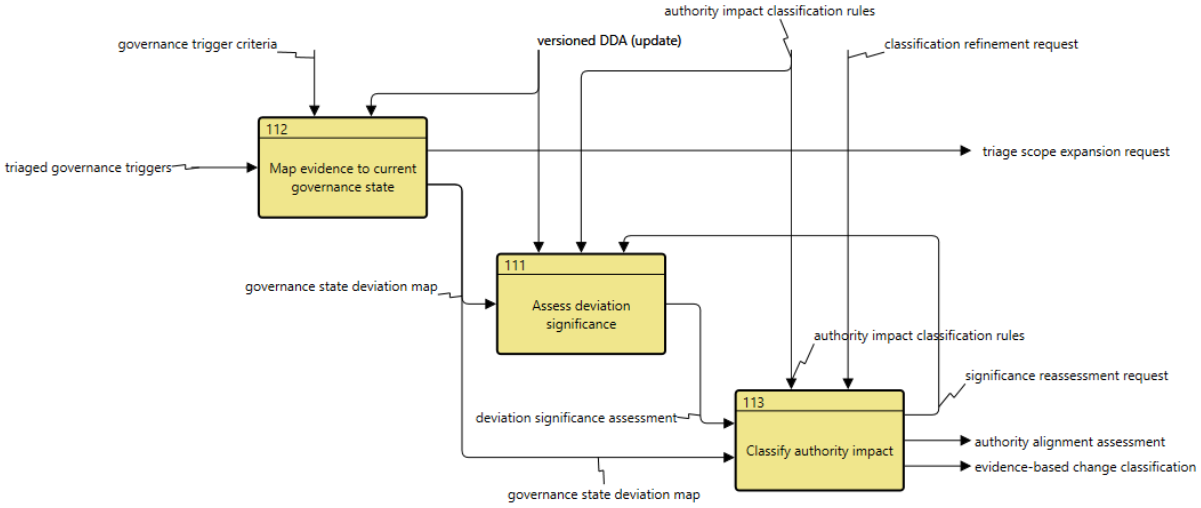

Figure 4.7: Functional decomposition of "Assess authority alignment" (F11). Three sub-functions map incoming evidence to the current governance state, assess the significance of observed deviations, and classify their impact on delegated authority.

On the basis of the authority alignment assessment, the architecture determines the appropriate revision pathway. This results in an authority review decision: a formal governance judgement indicating whether the current authority arrangement remains valid, requires refinement, or necessitates broader re-authorisation. The decision does not itself revise the specifications; rather, it provides the governance basis on which such revisions may be initiated through the specification functions. Finally, the outcome is formalised through a versioned update of the DDA, recording the trigger, evidence, assessment outcome, and any resulting changes. Lifecycle calibration therefore ensures that delegated authority remains explicitly governed and traceable over time, rather than informally evolving through operational use.

4.3.6 The Delegation Decision Artefact

The Delegation Decision Artefact (DDA) is the governance record of the DGA. It is not a separate decision mechanism, but the artefact through which the outputs of the DGA are preserved and versioned over time. Its function is to record what was authorised, on what basis, under which conditions, and how those decisions subsequently changed.

This makes the DDA a traceability artefact rather than a replacement for operational governance specifications. The Delegated Authority Specification and the Monitoring and Enforcement Configuration are used by surrounding governance and operational functions. The DDA records and justifies those artefacts so that delegated authority remains explicit and accountable across the lifecycle.

Table 4.3: Contents of the Delegation Decision Artefact (DDA).

DDA element	**Content**
Decision basis	Deployment intent, capability assumptions, relevant constraints, evidence basis, and approval conditions.
Authority Boundary Specification	The recorded boundary within which delegation was considered.
Delegated Authority Specification	The authorised action scope, including classifications, escalation conditions, and review status.
Monitoring and Enforcement Configuration	The governance-relevant signals, escalation logic, enforcement responses, and feasibility assumptions attached to the delegation.
Version history	The sequence of governance updates over time, including trigger source, evidence basis, review outcome, and resulting changes.

The DDA ensures that delegated authority does not remain implicit in system design or operational practice. It provides a structured record through which decisions can be reconstructed, reviewed, and revised over time. A concrete example of a completed DDA is provided in Chapter 5, where the artefact is applied in a deployment context.

Chapter 5

Evaluation of the DGA

This chapter evaluates the DGA as a design artefact. Given the limited maturity of enterprise-scale A-AI deployments, and given that the full architecture would only be instantiated at a later stage of organisational adoption, a live evaluation of operational effectiveness was beyond the scope of this research. The evaluation therefore focuses on what can credibly be assessed at this stage: whether the core governance logic underpinning the DGA can be meaningfully explored in a realistic deployment context, whether practitioners recognise the governance problem it addresses, and whether the surrounding organisational conditions required for its operation can be located in enterprise practice.

The evaluation proceeds through three streams. First, the underlying concept of delegated authority was examined against the A-AI deployment described by Lazăr [29], in order to assess whether this framing helps structure governance questions in a realistic enterprise case. Second, practitioner appraisal was conducted through a short pre-questionnaire and a structured discussion session with CGI Director Consulting Services (DCS) practitioners, in order to examine whether the governance problem and the relevance of a structured response are recognised in practice. Third, an organisational-fit analysis was conducted to assess whether the surrounding governance functions required by the DGA can be meaningfully located in the CGI context. Together, these evaluation activities provide evidence of conceptual applicability, practitioner recognisability, and organisational plausibility.

5.1 Applied Validation

The first evaluation stream examined whether the governance logic underlying the DGA helps make a realistic A-AI deployment more governable in conceptual terms. Rather than applying the full DGA as an operational architecture, the evaluation focused on the more foundational idea of *delegated authority* as a governance lens. This was appropriate because the full DGA is a more elaborate lifecycle architecture whose complete instantiation would depend on later-stage organisational adoption, explicit role allocation, and supporting governance routines.

For this purpose, the deployment described by Lazăr [29] was used as a realistic enterprise scenario. The case was selected because it provides a CGI-based intervention context in which agentic functionality is proposed but not yet fully operationalised, which aligns with the ex ante evaluation logic of the present research. The deployment comprises an A-AI-supported documentation workflow involving four agents: a parser agent, a tracker agent, a librarian agent, and an assistant agent, supported by semantic search. The intervention is framed as a socio-technical system and already identifies human oversight, workflow adaptation, and control over agent autonomy as important conditions for responsible deployment. The purpose of the evaluation was to examine how delegated authority clarifies what would need to be governed if such a deployment were to proceed.

The evaluation unfolded in two steps. First, an early overview of the present research was shared with the author of the intervention, who used it to reassess the proposed deployment and introduce additional governance-oriented elements. Second, the present research developed a further interpretation of the scenario using the fuller delegated-authority logic that later informed the DGA. The evaluation therefore does not claim that the completed DGA was fully applied to the case. Rather, it shows how the emerging governance logic informed the intervention iteratively and how further specification becomes visible as that logic is developed.

Using delegated authority as an analytical lens made several governance elements explicit that were only partly specified in the original intervention design. In particular, it required consideration of authority boundaries for each agent, forms of action requiring escalation or constraint, and the monitoring and review needed if delegated scope were to evolve over time.

Table 5.1 summarises this progression. It distinguishes between elements present in the original intervention design, additions introduced by Lazăr [29] after engagement with an early overview of the present work, and further governance specifications developed later in this research.

Table 5.1: Illustrative application of the DGA to the deployment described by Lazăr [29]. The table distinguishes between elements already present in the original intervention plan, governance-oriented additions introduced in Lazăr [29] after application of the DGA, and additional governance specifications developed in the present research.

Aspect	**Initially in Lazăr [29]**	**Added in Lazăr [29] after DGA use**	**Added in this research**
Agent scope	Four-agent workflow defined	Individual agent freedom introduced	Explicit authority boundary per agent
Control over action	Expected outcome gate for task assignment	Deployment / rollback / review controls added	Action classification into permitted, escalation-required, and non-permitted categories
Operational oversight	Tracker and Librarian support visibility	ADO lifecycle tagging and consultant rotation added	Monitoring scope, governance signals, and review triggers specified
Governance record	Implicit across intervention design	—	Versioned DDA structure for recording authority decisions and revisions

This evaluation did not demonstrate the operation of the full DGA. Its value lies in showing that the emerging governance logic of delegated authority can be used coherently to reinterpret a realistic deployment proposal, surface under-specified governance questions, and inform further refinement of the intervention. In particular, it highlights the need to distinguish capability from authority, to make authority conditions explicit, and to treat post-deployment review as part of the governance object rather than as an afterthought. These observations support the conceptual relevance of the governance problem to which the DGA responds.

5.2 Practitioner Appraisal

The second evaluation stream examined whether practitioners recognised the governance problem addressed by the DGA and regarded a structured governance response as relevant in an enterprise setting. This appraisal consisted of a short pre-questionnaire and a structured discussion session with CGI DCS practitioners. The questionnaire was distributed to all DCSs in Groningen, and the structured session was

held with three participants from that group. Due to time constraints, the session covered the full model only in abbreviated form and did not exhaust the original discussion protocol.

Given the limited participation in the session, this evaluation stream is interpreted as indicative rather than representative. Its purpose was not to establish consensus, but to examine how the governance problem and the proposed architectural response are received by practitioners working close to the governance–delivery interface.

The pre-questionnaire comprised five Likert-scale items and three open questions. Table 5.2 presents the quantitative responses.

Table 5.2: Pre-questionnaire quantitative responses (scale 1–5, $n = 6$).

Item	Question	Avg.	Range
Q1	Familiarity with agentic AI	4.8	4–5
Q2	Expected relevance within two years	3.8	3–5
Q3	Readiness of current processes to govern A-AI	2.7	2–3
Q4	Clarity of accountability for unapproved A-AI decisions	2.8	2–4
Q5	Sufficiency of internal knowledge sharing on A-AI governance	2.2	1–4

Although based on a small number of responses, the overall pattern is clear. Familiarity with agentic AI was high, and expected relevance within two years was also relatively high. By contrast, perceived readiness of current processes to govern what an A-AI system may and may not do, and to maintain that governance as the system evolves, was notably lower. Clarity of accountability for undesirable or unapproved A-AI decisions was similarly low, and internal knowledge sharing on A-AI governance received the lowest score overall.

The open responses and structured session added context to these scores. Participants highlighted uncertainty about what is and is not permitted under both CGI and client-specific constraints, data sovereignty concerns relating to data location and access, and the observation that many agentic AI use cases are not yet sufficiently concrete to have generated structured governance routines. In the discussion session, the governance problem addressed by the DGA was recognised as genuine and relevant, but participants also questioned whether the organisational preconditions required to operate a framework of this kind are currently in place. This was captured particularly clearly by one participant, who remarked at the close of the session that a governance framework of this kind "should have been created three years ago."

Given the limited number of participants, these findings should be interpreted as indicative rather than representative. Their evidential value lies in two points. First, the gap between relatively high perceived relevance and comparatively low perceived readiness suggests that the governance problem addressed by the DGA is recognised by practitioners as near-term and practically significant, while the organisational conditions required to respond to it are understood to be only partly in place. Second, the session reinforced a pattern already visible in the case study interviews: governance questions were frequently reframed as technical or operational questions, such as data sovereignty, runtime controls, and system architecture. These are important considerations, but within the logic of this research they function as constraints on delegation rather than as substitutes for governance specification. This suggests that the distinction between capability and authority, which is central to the DGA, is not yet fully stabilised in practice. The implications of these observations for the positioning of the DGA and for organisational governance readiness are discussed further in Chapter 6.

5.3 Organisational Fit and Readiness at CGI

The third evaluation stream examined whether the organisational functions required to operate the DGA can be meaningfully located in the CGI context, and what their current state reveals about governance readiness. Chapter 4 presented nine surrounding governance functions across three architectural layers (see Section **??**, Figures C.1–C.3), each with a generic component responsible for performing it. This section instantiates those functions in the CGI context by assessing, for each, whether the required component exists, partially exists, or is absent. Table 5.3 summarises the result.

Table 5.3: CGI readiness for operating the surrounding governance functions of the DGA.

Function	Component	Status	Current basis at CGI
Highest-level context: enterprise governance functions			
Define agentic AI strategic intent and governance policy	Agentic AI governance board	Partial	Governance councils, Office of the CTO, responsible AI principles
Manage agentic AI knowledge and practice	Agentic AI centre of excellence	Partial	AI communities, AI Exchange, AI delivery process, AI risk matrix
Manage agentic AI portfolio and resources	AI portfolio management	Partial	AI Executive Steering Committee, investment structures
Middle-level context: deployment lifecycle functions			
Develop agentic AI deployment proposal	Consulting / solution team	Partial	DCS consulting, opportunity framing practices
Design and build agentic AI solution	AI/ML engineering team	Partial	AI engineering and architecture capability
Deploy and operate agentic AI system	Operations / managed services team	Partial	Delivery and managed services operations
Lowest-level context: deployment governance functions			
Establish deployment governance specification	Delegation governance analyst	Absent	Relevant roles exist; no assigned accountability
Monitor runtime operations and manage incidents	AI operations monitoring team	Absent	Monitoring roles exist; not governance-aware
Validate and authorise deployment	Deployment governance authority	Absent	Approval structures exist; not for delegated authority

Here, "partial" denotes that related roles, processes, or structures exist at CGI, but not yet in the explicit, integrated, or routinely operationalised form required to support the DGA.

The pattern across the three layers is consistent, but the nature of the gap differs at each level.

At the *highest level* (see Figure C.1), the enterprise governance functions are partially in place. CGI has governance councils, an Office of the CTO, and RAI principles that provide enterprise-level direction for AI strategy and governance within the organisation's broader risk and compliance posture. However, the outputs of these bodies remain scattered and not easily found. They do not yet consolidate into visible, actionable governance rules and guidelines that downstream functions can routinely operate within. Governance-relevant knowledge exists through AI communities, the AI Exchange platform, and practice artefacts such as the AI delivery process and AI risk matrix, but this knowledge is distributed across platforms and individuals rather than integrated into a coherent, standardised governance knowledge base. Portfolio and resource oversight is supported by the AI Executive Steering Committee, but portfolio-level visibility into the governance status of individual A-AI deployments is not yet established.

At the *middle level* (see Figure C.2), the deployment lifecycle functions have the strongest organisational foundation. CGI has established consulting, engineering, and operational delivery capability, and the agentic

AI delivery process provides a documented structure for these functions. However, operationalisation of this process remains limited: awareness among practitioners is still low, and the process is not yet routinely followed in practice. Furthermore, the lifecycle functions in their current form do not include explicit delegation scope, authority specification, or governance-aware observability instrumentation as standard outputs.

At the *lowest level* (see Figure C.3), the deployment governance functions closest to the DGA system boundary are absent. This does not mean that CGI lacks the underlying organisational roles entirely. Consulting, operations, and approval roles all exist. What is missing is the specific accountability and authority required for delegated-authority governance: no role is currently responsible for translating deployment proposals into governance specifications, no function performs governance-aware monitoring of delegated authority in operation, and no authority reviews and approves delegated authority as a distinct governance object. The roles exist; the governance mandate does not.

5.4 What the Evaluation Shows

The evaluation shows that the governance problem to which the DGA responds is conceptually applicable, recognisable in practice, and organisationally plausible, even though the full architecture was not instantiated or tested in live operation.

In the deployment scenario, the lens of delegated authority made visible a set of governance questions that remained under-specified in the intervention design itself, particularly with regard to authority boundaries, escalation conditions, monitoring requirements, and review triggers. In the practitioner appraisal, the governance problem was recognised as relevant, while also indicating that the organisational conditions required to respond to it are only partly in place. In the organisational-fit analysis, the surrounding functions required by the DGA could be meaningfully located in the CGI context, but were shown to vary considerably in maturity.

These findings reveal a gradient of governance readiness: partial capability at the enterprise level, partially operationalised processes at the deployment lifecycle level, and absent or weakly defined accountability at the level of deployment governance. As governance moves from strategic intent towards the operational control of specific agentic systems, the supporting structures become less explicit and less consolidated. This pattern indicates that while the DGA is organisationally plausible, it also makes visible which governance capabilities remain implicit, fragmented, or absent.

The evaluation does not demonstrate live operational effectiveness, nor does it show the full application of the DGA as an instantiated governance architecture. What it does show is that the underlying governance object of delegated authority can be meaningfully identified in realistic deployment contexts, that practitioners recognise the relevance of the problem, and that the organisational conditions required for a fuller architectural response can be diagnosed in enterprise practice. On that basis, the evaluation supports the relevance and plausibility of the DGA while also clarifying the organisational work that would be required for its fuller adoption. The broader implications of this governance-readiness gradient are discussed in Chapter 6.

Chapter 6

Discussion

This chapter reflects on what the research contributes and why it matters. At its core, the study makes a simple but consequential shift: it treats delegated authority in A-AI as something that must be governed explicitly, rather than left implicit in technical design choices, workflow configurations, or local implementation practices.

On that basis, this chapter discusses the conceptual and architectural contributions of the DGA, interprets what the evaluation does and does not demonstrate, considers the broader implications of the governance-readiness findings, and addresses the main limitations of the study and directions for future research.

6.1 Conceptual Contribution

The conceptual contribution of this research lies in how it reframes the governance problem posed by A-AI. Rather than treating governance simply as an extension of risk management or compliance assurance, the research centres on a more specific question: how should organisations govern what an agentic system is permitted to do, and how should that permission evolve over time? This reframing has implications for how delegated authority is understood as a governance object, for how governance is distributed across the lifecycle, and for how the architecture relates to the organisational conditions required to operate it.

6.1.1 Delegated Authority as a Distinct Governance Object

The core conceptual contribution is the treatment of delegated authority as a governance object in its own right, distinct from capability, autonomy, and risk. Chapter 2 showed that existing AI governance frameworks are primarily organised around risk classification, compliance obligations, and assurance routines [16, 17, 25]. The challenge introduced by A-AI extends beyond risk management: it concerns delegation under conditions of incomplete observability, where organisations rely on systems that act within workflows without being able to reduce governance to technical capability alone [23, 26]. This makes a more fundamental governance question unavoidable: what is the system permitted to do, under which conditions does that permission hold, and how should it be revised as the system's effective scope of action changes? The distinction developed in Section 2.1 addresses this. Capability refers to what a system can technically do, and autonomy to how independently it operates. Neither determines what the system is allowed to do. Delegated authority instead captures what the organisation explicitly permits in a given deployment context.

These dimensions are related but not interchangeable. A system may be capable of tool use, coordination, or multi-step action without being authorised to perform those actions in practice. When this distinction is blurred, governance risks collapsing into technical description, with authority implicitly

embedded in design choices rather than established through explicit decisions. Taking delegated authority as the reference point shifts the focus of governance. Instead of centring on what systems can do and how risky that appears, governance must address what is allowed, what lies outside authorised scope, when escalation is required, and how to respond when practice begins to exceed initial assumptions.

This perspective also clarifies how the research relates to existing governance instruments. The EU AI Act defines binding requirements around risk classification and human oversight [8]. The National Institute of Standards and Technology (NIST) provides a voluntary lifecycle structure for managing AI risk [9]. ISO/IEC 42001 specifies management system requirements for organisational AI governance [10]. These frameworks are not deficient; rather, they address a different level of the problem. They define what organisations should pay attention to, but not how delegated authority should be explicitly bounded, monitored, and revised in agentic deployments. The DGA is therefore intended as a complementary layer that addresses this more specific governance challenge.

6.1.2 From Ex Ante Approval to Lifecycle Governance

A second conceptual contribution is the lifecycle view of delegated authority. The problem analysis showed that governance is often concentrated in pre-deployment stages, with much less structured attention to how authority evolves after deployment (Section 2.4, Figure 2.2). This is problematic because agentic systems are not static: their role can shift as tool access expands, workflows adapt, user reliance changes, or new dependencies emerge.

The distinction between authority-preserving and authority-altering change, introduced in Section 2.3, is central to this perspective. It formalises what is often left implicit: whether changes in system behaviour remain within authorised scope or constitute a governance-relevant expansion of authority. This contribution extends beyond the DGA. Any adaptive system operating under organisational constraints raises the question of when operational change becomes governance-significant. Making that threshold explicit remains largely unaddressed in current frameworks.

This lifecycle perspective also clarifies the relationship between the DGA and its organisational setting. The DGA does not operate in isolation, but depends on a surrounding governance environment to provide deployment intent, capability information, evidence, and policy constraints (see Appendix C). This dependence is not a limitation, but part of its value. The DGA structures delegation decisions where governance conditions are sufficiently mature, while making gaps visible where they are not. If authority boundaries cannot be clearly defined, if monitoring is not feasible, or if accountability remains under-specified, the architecture exposes these limitations rather than concealing them. The evaluation confirmed this dual role: the DGA not only supported structured decision-making on delegated authority, but also highlighted the organisational capabilities required for such governance to function in practice. This diagnostic function is therefore a key part of its contribution.

6.2 Architectural and Practical Contribution

Beyond the conceptual reframing, the research contributes a concrete governance architecture. Where existing RAI practice tends to leave a gap between high-level principles and operational governance decisions [16, 17], the DGA occupies that gap by translating the five structural governance gaps identified in Chapter 2 into a structured set of governance functions and artefacts. Two aspects of this architectural contribution warrant discussion: the move from governance principles to governance specification, and the role of the DDA as a mechanism for lifecycle traceability.

6.2.1 From Governance Principles to Governance Specification

The practical contribution of the DGA lies in making delegated authority explicit: bounded, classified, monitored, and recorded through defined artefacts. The use of systems engineering logic, specifically IDEF0 functional decomposition, provides structural rigour. It specifies what must be governed, how governance functions relate, what inputs they require, and which artefacts they produce. This does not make the architecture inherently more correct, but it makes it explicit, traceable, and revisable. Crucially, the design accommodates change by construction. Governance functions are defined in relation to controls such as policy constraints and regulatory requirements. When these controls change, for example due to new or updated legislation, the affected functions and artefacts can be systematically identified and updated. This makes the architecture responsive to the changing A-AI landscape, where use cases, technical capabilities, and external requirements continue to evolve.

A key design choice is the separation between governance specification and organisational authorisation. The DGA does not itself grant approval. Instead, it structures the basis on which approval can be made by defining authority boundaries, delegation conditions, escalation points, monitoring requirements, and review logic. The act of authorisation remains with the surrounding governance function. This separation matters because it makes the architecture more adaptable: it can be integrated into existing organisational approval processes and used at different lifecycle stages, including phase gates, without presupposing a particular delivery methodology or governance model.

6.2.2 The Role of the DDA and Lifecycle Traceability

The DDA plays a central role in enabling lifecycle traceability. Without a structured record, decisions about delegated authority risk becoming dispersed across documents, meeting notes, configuration choices, and individual memory, making later review difficult. The DDA addresses this by consolidating core governance artefacts into a versioned record that captures what was authorised, on what basis, under which conditions, and how those decisions evolved over time (Table 4.3).

This function becomes more important as A-AI deployments mature. As systems are reconfigured, extended with new tools, or embedded more deeply in workflows, the original intent and authorised scope can become difficult to reconstruct. This weakens accountability and makes it harder to assess whether changes remain within scope or require renewed governance attention. The DDA provides the structural basis to address this risk, even if its effectiveness depends on disciplined use in practice. Without such traceability, delegated authority may expand in operation while the basis for governing it becomes increasingly difficult to recover.

6.3 Interpreting the Evaluation

Chapter 5 presented three complementary evaluation streams: applied evaluation, practitioner validation, and organisational-fit analysis. Taken together, these do not demonstrate live operational effectiveness. They do, however, provide evidence on three distinct questions: whether the governance logic underlying the DGA can be used coherently in a realistic deployment context, whether practitioners recognise the governance problem as relevant, and whether the organisational conditions required to operate the architecture are currently in place. This section interprets those findings, beginning with applicability before turning to the more structural results concerning organisational readiness.

6.3.1 Applicability and Recognisability

The applied evaluation demonstrates that the governance logic underlying the DGA can be meaningfully used in a realistic deployment context and can generate governance outputs that were only partly explicit

in the original intervention design [29]. It required authority boundaries to be considered, actions to be classified, escalation conditions to be specified, and monitoring requirements to be articulated. These elements were not wholly absent beforehand, but they had not been formalised as governance artefacts. The delegated-authority lens therefore adds value by making otherwise implicit governance assumptions visible and by recasting them as matters of governance rather than of design convenience.

The practitioner validation supports the relevance of this contribution. Participants reported high familiarity with agentic AI and expected it to become increasingly relevant in the near future, while rating current governance processes, accountability structures, and internal knowledge sharing much lower. This suggests that the problem addressed by the architecture is recognisable in practice even where governance arrangements remain underdeveloped. At the same time, the evaluation also showed that governance questions were often reframed as technical or implementation issues, such as data sovereignty, runtime controls, or system architecture. Those are important issues, but within the logic of the DGA they function as constraints on delegation rather than as substitutes for governance specification. This tendency therefore reinforces the central claim of the research: delegated authority is not yet consistently recognised in practice as a distinct governance object.

These findings should nonetheless be interpreted cautiously. The applied evaluation did not demonstrate the operation of the full DGA as a completed lifecycle architecture. Rather, it showed that the emerging delegated-authority logic underlying the architecture could be used coherently to reinterpret a realistic deployment proposal and to surface governance questions that remained under-specified in the intervention design itself. Likewise, practitioner recognition of the problem does not demonstrate that the architecture is sufficient in operational use. What the evaluation does support is the more modest claim that the governance logic is intelligible, applicable, and timely.

6.3.2 Governance Readiness as a Structural Finding

The organisational-fit analysis adds a third perspective to the evaluation by examining whether the governance functions required by the DGA can be located in the CGI context. Considered alongside the applied evaluation and practitioner validation, this points to a broader structural finding: governance readiness for agentic AI is not only uneven, but appears systematically weaker at the point where delegated authority must actually be specified, monitored, and revised.

The CGI case suggests a layered pattern. Enterprise-level policy and oversight structures are present, and deployment lifecycle processes exist in documented form. However, governance becomes less mature as it moves closer to the operational control of delegated authority. In particular, accountability for specifying, monitoring, and re-authorising delegated authority remains comparatively weak. This pattern aligns with the practitioner findings: familiarity and perceived future relevance scored relatively high, whereas clarity of accountability, governance readiness, and knowledge sharing scored substantially lower. These findings suggest a mismatch between the growing practical importance of agentic AI and the maturity of the governance infrastructure needed to manage delegated authority explicitly.

This should not be interpreted simply as a local weakness of CGI. The organisation was selected precisely because it represents a relatively mature governance setting and an organisation actively engaging with A-AI. If the prerequisites for governing delegated authority remain only partly established in such a context, it is plausible that similar gaps exist more broadly. The issue is therefore not merely that some organisations have not yet caught up. More fundamentally, governance maturity developed around advisory and generative AI does not automatically extend to agentic systems. When systems move from informing human decision-making to acting within workflows, the governance problem changes. Instruments and processes designed for the former may therefore create a false sense of readiness for the latter.

This finding also shifts attention from the DGA itself to the conditions required to operate it. As specified through the IDEF0 decomposition (Appendix C), the architecture depends on a surrounding

governance environment that can provide a coherent deployment intent, a system capability profile, governance policies and rules, observability constraints, operational delegation requirements, and clearly mandated governance roles (Section **??**). These are not optional features, but structural prerequisites for governing delegated authority.

The practical implication is that the DGA can only be operationalised where the surrounding governance environment is able to supply these inputs and controls. Where they are absent, the architecture cannot substitute for them. At the same time, this gives the DGA an important diagnostic role: it makes visible what must be established before responsible agentic deployment can proceed at scale. One of the central findings of this study is therefore not only about the architecture itself, but about the current state of enterprise governance for A-AI: the DGA specifies how delegated authority can be governed, while simultaneously revealing that many organisations may not yet be fully equipped to do so.

6.4 Limitations

The claims made in this research should be read in light of several limitations concerning the empirical base, the form of evaluation, the scope of the artefact, and the broader governance problem to which it relates.

6.4.1 Empirical Scope and Evidential Base

The empirical base of the study is bounded. The problem investigation relies on a single case study within CGI. Although the case was purposefully selected and represents a relatively governance-mature organisation, the findings remain grounded in one context. The governance gaps identified are argued to be structural, but that claim has not been tested across multiple organisations or sectors. In addition, the research was conducted at an early stage of A-AI adoption. Most participants had substantial AI experience, but limited exposure to mature agentic deployments, meaning that the analysis concerns governance of an emerging and rapidly evolving phenomenon.

This limitation follows directly from the methodological strategy of the research. The study used a single embedded case to support problem structuring, empirical analysis, artefact design, and ex ante evaluation. This made it possible to develop a contextually grounded and design-relevant architecture, but it also means that the findings should be understood as analytically generalisable rather than statistically representative.

The empirical material was also shaped by practical constraints. Access to participants and relevant organisational artefacts developed gradually, and documentation was not always readily accessible. Confidentiality constraints limited access to internal materials. As a result, the analysis relied substantially on information shared through interviews and demonstrations rather than on systematic access to the full documentary basis of governance practice. This provided valuable insight into how governance is enacted in practice, but it limits the completeness and independent verifiability of the empirical evidence.

6.4.2 Evaluation Limits

The evaluation is ex ante. The DGA has been shown to be conceptually applicable and perceived as relevant, but it has not been tested in sustained operational use. The applied evaluation is best understood as an application of the emerging delegated-authority logic rather than as an independent test of the completed architecture, and the practitioner validation involved a small sample ($n = 6$ questionnaire, $n = 3$ session).

Accordingly, no evidence is yet available on how the DDA performs over multiple governance cycles, whether the review logic remains workable under operational pressure, or whether the governance overhead

introduced by the architecture is proportionate to its benefits. The evaluation therefore supports the plausibility and perceived usefulness of the artefact, but not its long-term organisational effectiveness.

6.4.3 Architectural Scope Limits

The DGA is a governance architecture rather than a complete organisational solution. It specifies what should be governed and how governance decisions should be structured, but it does not itself implement monitoring, enforcement, or approval processes. Its effectiveness therefore depends on the surrounding governance and operational infrastructure, which, as the evaluation indicates, may not yet be fully established in practice.

In addition, the current design assumes a single-deployment scope. It does not address in detail how delegated authority should be governed across interacting, interdependent, or dynamically orchestrated agentic systems. As such systems become more common, governance may need to extend beyond the level of individual deployments towards portfolios or networks of delegated authority.

6.4.4 Limits of Problem Scope

Finally, the research deliberately focuses on delegated authority as a specific governance problem. The empirical work surfaced broader concerns that fall outside this scope but remain relevant to responsible A-AI deployment, including trust in autonomous systems, AI sovereignty, the ethical implications of replacing human judgement, bias, workforce effects, cultural variation, and cross-jurisdictional governance challenges.

These issues indicate that delegated authority is only one dimension of a wider governance problem. The contribution of this research is therefore necessarily partial: it addresses a governance gap that becomes especially important in agentic settings, but it does not claim to provide a comprehensive model of enterprise governance for A-AI as a whole.

6.5 Future Research

The findings of this research point to several directions for future research. The most immediate need is to move from architectural specification towards empirical and organisational operationalisation. At present, the DGA clarifies what must be governed and through which functions and artefacts, but it does not yet show in detail how those functions can be embedded in day-to-day governance practice across different organisational settings.

6.5.1 Operationalising the Architecture in Organisational Practice

A first priority is therefore implementation-oriented research. The evaluation in Chapter 5 provided an initial indication of applicability, but it did not yet examine sustained operational use. Future research should study how the DGA can be embedded into concrete governance routines through templates, decision protocols, role definitions, approval checkpoints, and maintenance procedures for the DDA. This includes examining where the architecture fits within existing delivery, risk, compliance, and approval processes, which governance actors are best positioned to operate particular functions, and what level of effort is realistic for maintaining lifecycle review over time.

Such work should also distinguish between different organisational adoption pathways. In some settings, agentic AI may be introduced incrementally within existing delivery and governance structures, so that the DGA functions as a complementary governance layer. In other settings, more agentic forms of AI may alter workflows, decision rights, and accountability structures more fundamentally. In that case, governance of delegated authority cannot simply be added to existing arrangements, but must be designed together

with broader organisational change. Future research should therefore examine under what conditions each pathway applies and how the architecture should be adapted accordingly.

6.5.2 Comparative and Longitudinal Case Study Research

A second direction is comparative case study research. Because this study is grounded in a single embedded case at CGI Netherlands, future research should examine whether the governance gaps identified here also appear in other contexts and, if so, in what form. A useful next step would be a comparative design that deliberately varies both governance maturity and contextual specificity. One set of cases could examine organisations similar to CGI, such as large professional-services or technology firms with relatively mature RAI structures, in order to test whether the readiness gaps identified here persist even in comparatively advanced governance environments. A second set could examine deployment-specific cases in more highly regulated domains, such as financial services, healthcare, energy, or government, where acceptable delegation, auditability requirements, and regulatory exposure are more tightly defined by sector-specific obligations. This would complement the present study, which intentionally began from a general enterprise governance architecture in a cross-sector setting rather than from one domain-specific deployment context. A third set could examine organisations that are actively experimenting with agentic systems but have less formal governance infrastructure, to assess whether the architecture remains usable under lower-maturity conditions or requires simplification.

Longitudinal research would be equally important. The current study evaluates the DGA ex ante, but does not observe how delegated authority is reviewed and revised across repeated governance cycles. Future work should therefore follow one or more deployments over time, ideally from early proposal through operational change and later re-evaluation. This would make it possible to assess whether the review logic remains workable under real organisational pressure, how often revision is actually triggered, whether the DDA remains maintainable, and whether the governance overhead remains proportionate to the value of tighter control over delegated authority.

6.5.3 Organisational and Socio-Technical Conditions of Delegation

A third direction concerns the wider organisational and socio-technical conditions under which delegated authority becomes governable. The empirical work in this study repeatedly pointed beyond formal governance structures towards issues such as trust in autonomous systems, organisational learning, the redistribution of work between humans and agentic systems, the ethical implications of replacing or bypassing human judgement, client demand for increasingly autonomous solutions, and questions of sovereignty and cross-jurisdictional governance. These concerns shape whether delegated authority is acceptable in practice, how governance is interpreted by practitioners, and how much authority organisations are willing to delegate in the first place.

Future research should thus examine how delegated authority interacts with changing work practices and organisational expectations. This includes questions such as how trust in A-AI systems is formed or undermined over time, how employees experience the transfer of judgement or discretion to AI systems, how organisations learn from incidents and near misses, and how client expectations influence the pace and form of agentic adoption. It would also be valuable to study whether delegated authority should be treated as one governance layer within a broader socio-technical governance architecture, or whether it can serve as an organising core around which adjacent concerns such as ethics, learning, workforce impact, and sovereignty are structured.

6.5.4 Delegated Authority in Broader Socio-Technical Context

A third direction concerns the wider socio-technical context within which delegated authority is embedded. The empirical work surfaced issues such as trust in autonomous systems, data sovereignty, organisational learning, ethical concerns, sustainability, and cross-jurisdictional governance. These dimensions fell outside the scope of the present study, but they are not peripheral to the responsible deployment of agentic AI.

Future research should therefore examine how such concerns interact with delegated authority and whether they require dedicated governance mechanisms of their own or can be incorporated as extensions of the DGA. This would help clarify whether delegated authority should be treated as one governance layer within a broader socio-technical architecture, or whether it can serve as an organising core around which other governance concerns are structured.

6.5.5 Delegated Authority Across Interacting Agents

Finally, the governance of delegated authority across interacting agents warrants dedicated attention. The current architecture assumes a single-deployment scope. As enterprise A-AI systems become more distributed, authority may no longer be confined to one bounded system, but may emerge across chains of interacting agents, tools, models, and workflows. In such settings, the central governance problem is extended to how delegated authority is produced, transferred, constrained, and reviewed across multiple connected actors.

Future research should therefore broaden the architectural logic of the DGA to multi-agent and cross-system settings. This would require examining how authority boundaries should be specified when action is distributed, how escalation and review should operate across system boundaries, how accountability should be assigned when harmful outcomes emerge from interaction rather than from a single decision point, and how governance records such as the DDA might need to evolve to capture relational and cascading forms of delegated authority. As A-AI systems mature, this is likely to become one of the most important areas for further development.

Chapter 7

Conclusion

This research asked: *What enterprise governance architecture is needed to specify, monitor, and revise delegated authority in A-AI systems across their lifecycle?* The research argues that enterprise governance for A-AI requires an architecture that treats delegated authority as an explicit lifecycle governance object. Such an architecture must define what authority may be delegated, under which conditions that delegation holds, how it is monitored in operation, and when it must be reviewed or revised. The DGA was designed to meet this need.

The contribution of the research is therefore not another general AI governance framework, but a more specific conceptual and architectural response to the governance challenge introduced by agentic systems. It shows that delegated authority should be distinguished from capability and autonomy, and that governance must extend beyond one-off approval towards explicit lifecycle management of authorised scope. When mapped onto the CGI context, the architecture also functions as a diagnostic instrument, revealing that governance readiness is systematically weakest at the point where delegated authority must be made operationally explicit.

The evaluation does not demonstrate sustained operational effectiveness, but it does show that the underlying governance logic is conceptually applicable, recognisable to practitioners, and diagnostically useful in revealing the organisational conditions required for its operation. The broader implication is that as organisations move towards workflow-embedded A-AI, responsible governance will depend increasingly on making delegated authority explicit, reviewable, and governable over time.

Bibliography

[1] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey," *IEEe Access*, 2025.

[2] Anthropic, *Let Claude use your computer in Cowork*, Support documentation, Accessed: 2026-03-25, 2026. [Online]. Available: https://support.claude.com/en/articles/14128542-let-claude-use-your-computer-in-cowork.

[3] Microsoft, *Secure agentic AI end-to-end*, Blog post, Agent 365 generally available from May 2026., Mar. 2026. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2026/03/20/secure-agentic-ai-end-to-end/.

[4] OpenClaw, *OpenClaw: Local-first AI agent framework*, Product website, 2026. [Online]. Available: https://openclaw.ai/.

[5] M. Rodriguez, *Pick your agent: Use Claude and Codex on Agent HQ*, The GitHub Blog, Feb. 2026. [Online]. Available: https://github.blog/news-insights/company-news/pick-your-agent-use-claude-and-codex-on-agent-hq/.

[6] Gartner, Inc., "Gartner predicts over 40% of agentic AI projects will be canceled by end of 2027," Gartner, Press Release, Jun. 2025, Predicts 40% of enterprise software applications will have embedded AI agents by end of 2026. Accessed: 2026-03-25. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027.

[7] McKinsey and Company, *Elevating board governance through AI posture and archetypes*, McKinsey Insights, Based on interviews with directors from 75 boards and the McKinsey Global Survey on the state of AI. Accessed: 2026-03-25, Dec. 2025. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-technology/our-insights/the-ai-reckoning-how-boards-can-evolve.

[8] European Parliament and Council of the European Union. "Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act)." OJ L 2024/1689. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

[9] E. Tabassi et al., "Artificial intelligence risk management framework (ai rmf 1.0)," National Institute of Standards and Technology, Tech. Rep. NIST AI 100-1, Jan. 2023, U.S. Department of Commerce. DOI: 10.6028/NIST.AI.100-1. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1.

[10] *Iso/iec 42001:2023 artificial intelligence management system*, International Organization for Standardization, 2023.

[11] A. Bandi, B. Kongari, R. Naguru, S. Pasnoor, and S. V. Vilipala, "The rise of agentic ai: A review of definitions, frameworks, architectures, applications, evaluation metrics, and challenges," *Future Internet*, vol. 17, no. 9, p. 404, 2025.

[12] Y. Shavit et al., "Practices for governing agentic ai systems," *Research Paper, OpenAI*, 2023.

[13] E. Miehling et al., "Agentic ai needs a systems theory," *arXiv preprint arXiv:2503.00237*, 2025.

[14] R. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.

[15] R. L. Ackoff, "The art and science of mess management," *Interfaces*, vol. 11, no. 1, pp. 20–26, 1981.

[16] E. Papagiannidis, P. Mikalef, and K. Conboy, "Responsible artificial intelligence governance: A review and research framework," *The Journal of Strategic Information Systems*, vol. 34, no. 2, p. 101 885, 2025.

[17] A. Batool, D. Zowghi, and M. Bano, "Ai governance: A systematic literature review," *AI and Ethics*, pp. 1–15, 2025.

[18] I. D. Raji et al., "Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 33–44.

[19] L. Floridi and J. Cowls, "A unified framework of five principles for ai in society," *Machine learning and the city: Applications in architecture and urban design*, pp. 535–545, 2022.

[20] A. K. Pati, "Agentic ai: A comprehensive survey of technologies, applications, and societal implications," *IEEE Access*, 2025.

[21] T. Raheem and G. Hossain, "Agentic ai systems: Opportunities, challenges, and trustworthiness," in *2025 IEEE International Conference on Electro Information Technology (eIT)*, IEEE, 2025, pp. 618–624.

[22] R. Sapkota, K. I. Roumeliotis, and M. Karkee, "Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges," *Information Fusion*, p. 103 599, 2025.

[23] A. Baird and L. M. Maruping, "The next generation of research on is use: A theoretical framework of delegation to and from agentic is artifacts," *MIS quarterly*, vol. 45, no. 1, pp. 315–341, 2021.

[24] N. Kolt, M. Shur-Ofry, and R. Cohen, "Lessons from complexity theory for ai governance," *arXiv preprint arXiv:2502.00012*, 2025.

[25] S. Joshi, "Framework for government policy on agentic and generative ai: Governance, regulation, and risk management," *Regulation, and Risk Management (August 01, 2025)*, 2025.

[26] K. M. Eisenhardt, "Agency theory: An assessment and review," *Academy of management review*, vol. 14, no. 1, pp. 57–74, 1989.

[27] Google LLC, "Responsible ai progress report," Google, Tech. Rep., Feb. 2025.

[28] A. Khalili et al., "Knowledge-enriched agentic ai workflows: Intelligent and responsible task execution for business growth," Deloitte The Netherlands, The Netherlands, Whitepaper, 2025. [Online]. Available: https://www.deloitte.com/nl/en/services/risk-advisory/perspectives/responsible-enterprise-decisions-knowledge-enriched-ai.html.

[29] A. Lazăr, "A systems approach to ai-supported workflow efficiency at cgi," Master's Design Project, MSc Industrial Engineering and Management, Rijksuniversiteit Groningen, Feb. 2026.

[30] R. K. Yin, *Case study research and applications*. Sage Thousand Oaks, CA, 2018, vol. 6.

[31] C. Karlsson et al., *Research methods for operations management*. Routledge New York, 2016, vol. 2.

[32] P. Checkland, "Systems thinking, systems practice," 1981.

[33] B. Bergvall-Kåreborn, A. Mirijamdotter, and A. Basden, "Basic principles of ssm modeling: An examination of catwoe from a soft perspective," *Systemic Practice and Action Research*, vol. 17, no. 2, pp. 55–73, 2004.

[34] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[35] D. M. Buede, *The Engineering Design of Systems: Models and Methods*, 2nd. Wiley, 2009, ISBN: 9780470164020, 0470164026.

Appendix A

Interview and Stakeholder Analysis

This appendix documents the empirical material collected during the problem investigation phase. It provides the participant overview, interview protocol, coding structure, and representative evidence that underpin the analysis in Chapter 2. The methodological rationale for the case study design, participant selection, and analytical approach is explained in Sections 3.1 and 3.3.

Three forms of empirical engagement informed the problem investigation. Semi-structured interviews formed the primary coded dataset, comprising eleven sessions with thirteen practitioners involved in AI delivery, governance, architecture, strategy, responsible AI, and compliance. In addition, internal knowledge sessions and webinars provided contextual understanding of governance developments and implementation patterns, whilst executive and strategy discussions helped situate CGI's broader governance and strategic context. Only the formal interviews were coded systematically; the remaining material was used for contextual interpretation.

Table A.1 summarises the interviews and contextual sessions[1].

[1]All interview participants are anonymised. Roles and contexts are described at a level of abstraction intended to preserve analytical value whilst protecting confidentiality.

Table A.1: Overview of interview sessions and contextual knowledge sessions (anonymised)

ID	Role Category	Context	Date
Formal Interviews (Primary coded dataset)			
I1	Conversational AI delivery	Practitioner operating a live, regulated customer-facing deployment with increasing agentic characteristics; used as a comparative external case for translation, oversight, and autonomy-boundary issues	12 Dec 2025
I2	Global AI governance and applied AI leadership (pair)	Governance advisory, responsible AI, internal governance tooling, and applied AI enablement within CGI's global technology leadership	16 Jan 2026
I3	Global governance / risk tooling lead	Design and use of CGI's AI risk metrics tooling; mitigation logic, review responsibilities, and treatment of risk in delivery practice	19 Jan 2026
I4	Enterprise architect / consulting expert	Organisational incentives, architecture constraints, knowledge sharing, and barriers to internal reuse and operationalisation of AI assets	20 Jan 2026
I5	Business consulting and industry leadership	Strategic scaling of AI, operating model design, and the relation between consulting assets, industry context, and enterprise transformation	21 Jan 2026
I6	User-centred design and responsible AI leadership (pair)	Risk assessment, governance in project discovery, cultural and user considerations, and the distinction between enterprise governance and agent-level governance	23 Jan 2026
I7	AI strategy / governance practitioner	Early validation of an initial framework draft; emphasis on governance as umbrella, use-case assessment, risk mitigation, and staged scaling	23 Jan 2026
I8	SBU emerging technologies and AI leadership	Strategic-operational perspective on local AI maturity, use of risk tooling, internal approval processes, and barriers to experimentation	27 Jan 2026
I9	Responsible AI / ethical technology leadership	Internal implementation of CGI's responsible AI framework, weak process embedding, access issues, and divergence between formal governance design and operational use	29 Jan 2026
I10	Senior consultant / client delivery	Client-side adoption dynamics, local AI strategy formation, productionisation barriers, and investment constraints in delivery practice	04 Feb 2026
I11	IP leadership	Relationship between AI solutions, internal investment, risk, feedback from delivery, and organisational preconditions for scaling	16 Feb 2026
Knowledge Sessions / Webinars (Contextual only)			
W1	Global AI strategy / CoE leadership	CGI webinar on analytics and AI operating models, readiness, and central coordination structures	22 May 2025
W2	Cloud and regulated AI specialists	CGI webinar presenting regulated sector case studies and discussing oversight, governance, and deployment constraints	21 Aug 2025
W3	Executive technology leadership	Global strategy presentation covering governance bodies, strategic intent, and cross-SBU coordination	20 Nov 2025

A.1 Interview Protocol

The interviews were semi-structured and exploratory. Although the emphasis varied depending on the role and expertise of the participant, discussion typically covered the following areas:

1. **Current governance practice:** how governance is initiated, who is involved, where decisions are made, and how risks are assessed.
2. **Operational execution:** whether governance mechanisms are experienced as enabling, burdensome, bypassed, or weakly embedded in delivery practice.
3. **Authority and accountability:** how system scope is defined, who decides what the system may do, and how responsibility is handled when systems behave unexpectedly.
4. **Translation from principles to practice:** how high-level governance ideas, such as trustworthiness, oversight, or compliance, are converted into operational constraints, technical choices, and project requirements.
5. **System evolution and scaling:** how systems are extended, how capabilities are increased over time, and whether formal reassessment mechanisms exist.
6. **Oversight and intervention:** how human oversight is positioned, what intervention criteria exist, and whether review or escalation pathways are formalised.

These topics were derived in part from the provisional SSM diagnosis (Appendix B) and in part from the developing literature review. The interviews served both to ground the problem context empirically and to refine the evolving problem formulation.

A.2 Interview Analysis

The interviews were analysed using a theoretically informed deductive thematic analysis approach [34], guided by the governance tensions identified through the literature review and the SSM-informed problem diagnosis. Six themes were defined prior to coding, based on the analytical concepts that had emerged from the literature and from provisional problem structuring. Each theme represented a governance tension that the literature and early SSM work suggested should be present if the developing problem diagnosis was valid. The coding then examined whether, and how, these tensions were recognisable in the empirical material.

Table A.2 defines the six themes and indicates which governance gap each theme contributed to. The gap column should be read as retrospective traceability: it records which gap each theme ultimately informed, rather than a mapping that was fixed before the analysis began.

Table A.3 presents the binary coding matrix for the eleven formal interviews. A value of 1 indicates that explicit and defensible transcript evidence for the theme was identified in that interview; 0 indicates the absence of sufficiently direct evidence.

Several observations follow from the matrix. T1 (Artefact–Practice Gap) and T3 (Translation Gap) have the broadest coverage, each appearing in 7 of 11 interviews. This suggests that weak embedding of governance artefacts and difficulty in converting principles into operational rules are widely experienced across roles and organisational levels. T6 (Autonomy Boundary Ambiguity) appears in 6 of 11 interviews, indicating that uncertainty about where system authority may begin is also a broadly recognisable concern.

T4 (Incremental Expansion without Structured Reassessment) and T5 (Oversight Sustainability Tension) have comparatively narrow coverage, each appearing in only 2 of 11 interviews. This does not necessarily indicate that these issues are unimportant. Rather, it suggests that they are less institutionally salient in current practice than weak process embedding and translation difficulty. The organisation more

Table A.2: Deductive theme definitions and relation to governance gaps

ID	Theme	Definition	Gap(s)
T1	Artefact–Practice Gap	Formal governance artefacts, tools, or processes exist but are weakly embedded, poorly understood, bypassed, hard to access, or treated as administrative burden in delivery practice.	G1, G4
T2	Strategic Fragmentation and Scaling Tension	Strategic ambition regarding AI exists, but ownership, investment, cross-unit coordination, reuse, or scaling structures are fragmented or misaligned.	G4, G5
T3	Translation Gap	Difficulty converting high-level governance principles, legal requirements, or risk categories into concrete technical constraints, procedural rules, or design requirements.	G1, G2
T4	Incremental Expansion without Structured Reassessment	Recognition that capabilities, scope, or autonomy may increase over time, but without clear criteria, triggers, or formal mechanisms for cumulative reassessment and re-authorisation.	G3, G4
T5	Oversight Sustainability Tension	Reliance on human oversight is emphasised, but participants question its quality, feasibility, intervention basis, scalability, or long-term sustainability.	G2, G3
T6	Autonomy Boundary Ambiguity	Uncertainty about where human authority should end, where agent authority may begin, what requires escalation, and how autonomy progression should be governed.	G1, G3, G5

Table A.3: Theme–participant matrix (deductive coding, I1–I11)

Interview	T1	T2	T3	T4	T5	T6	Total
I1	0	0	1	0	1	1	3
I2	1	0	1	0	0	1	3
I3	1	0	1	1	0	1	4
I4	1	1	1	0	0	0	3
I5	0	1	0	0	0	0	1
I6	1	0	1	0	1	1	4
I7	0	0	1	0	0	1	2
I8	1	1	0	0	0	1	3
I9	1	1	1	0	0	0	3
I10	1	1	0	0	0	0	2
I11	0	1	0	1	0	1	3
Coverage	**7**	**5**	**7**	**2**	**2**	**6**	

readily perceives the friction of using governance artefacts than the need to govern authority evolution over time, a pattern consistent with the finding in Chapter 2 that governance is concentrated in early lifecycle stages.

The matrix provides a structured overview of thematic spread, but it necessarily abstracts from differences in depth and emphasis across interviews. Some themes were mentioned only briefly, whereas others were discussed at length as structural organisational barriers. The representative excerpts in Table A.4 illustrate how each theme manifested in the interview material.

Table A.4: Representative transcript evidence for the deductive themes

Theme	**Interview(s)**	**Representative evidence excerpt**
T1	I9	"The framework is there, it is solid, but it is not being used because people do not know it or do not want to." This excerpt illustrates the distinction between the formal existence of governance artefacts and their weak operational embedding.
T2	I4, I10	I4 described internal competition around frameworks and reuse, noting that when knowledge is treated as intellectual property rather than shared organisational capability, local teams are incentivised to build their own versions instead of reusing existing assets. I10 similarly highlighted fragmentation across units and weak collaboration around AI assets and delivery.
T3	I1	"People need to define the rules they want to give the bot. That is hard, because people do not really know that themselves." This reflects the difficulty of translating broad governance expectations into concrete behavioural rules and design constraints.
T4	I7, I11	I7 described implementation as a staged process in which lower-risk elements are introduced first and broader functionality is deferred until later. I11 likewise emphasised the absence of a clear overview of how learning from implementation feeds back into broader governance and strategy. Together these excerpts indicate incremental expansion without clearly formalised reassessment logic.
T5	I1	"On what basis do you intervene in the system?" This captures the concern that human oversight may be formally assigned without clear epistemic or operational grounds for meaningful intervention.
T6	I6, I1	I6 noted that governance requirements directly shape what kind of technical architecture is acceptable, for example whether only retrieval-based, closed-ended interactions are permissible rather than more autonomous agentic forms. I1 similarly stressed that someone must still decide "about autonomy and when something can or cannot happen". These excerpts illustrate ambiguity around where system authority may begin and where human authority must remain.

The themes and excerpts documented here provided the empirical basis for the analysis in Chapter 2. In the chapter body, these observations are interpreted in relation to the literature in order to develop the five structural governance gaps (G1–G5) that motivate the design of the DGA.

Appendix B

Soft Systems Methodology Artefacts

This appendix presents the SSM artefacts developed during problem investigation. Section 3.3 explains how SSM was used methodologically and what it contributed to the study. The artefacts presented here are interpretive rather than prescriptive: they structure the governance problem situation rather than the design response. The normative design response is developed through the governance gaps in Chapter 2 and the architecture in Chapter 4.

In this study, SSM was used to organise the governance problem situation, guide empirical inquiry, support iterative return to the literature, and narrow the study towards the governance of delegated authority in agentic AI systems. This appendix presents the resulting artefacts: the structured problem situation, the CATWOE analysis, and the root definition that informed the subsequent problem analysis.

B.1 Structured Problem Situation, CATWOE, and Root Definition

The rich-picture stage focused on understanding the problem situation as experienced across organisational levels. Four recurring insights emerged from iterating between the literature, early conversations, and the subsequent interview material.

First, **authority was often implicit rather than explicit**. Governance principles, risk controls, and review procedures existed, but what an agentic system was actually permitted to do was often embedded in technical design choices, workflow configuration, and local delivery decisions rather than defined as a distinct governance object. Second, **translation from principle to practice was weak**. High-level commitments such as accountability, transparency, oversight, and safety were widely recognised, but their conversion into operational constraints, escalation logic, and system-specific decision rules remained difficult. Third, **system evolution outpaced formal governance review**. Interviewees repeatedly described staged deployment and gradual capability expansion, but without equally clear criteria for when cumulative change should trigger renewed governance attention. Fourth, **accountability became less clear as systems evolved**. Responsibility for design, deployment, monitoring, and change was distributed across multiple actors, making it difficult to identify who remained responsible for the effective scope of delegation over time.

These insights were then structured further through CATWOE analysis. The purpose was not to produce a stakeholder map for its own sake, but to clarify what transformation a governance system would need to achieve if it were to address the problem coherently.

Customers Organisations deploying A-AI under regulatory and enterprise risk constraints, as well as the users, affected stakeholders, and regulators whose expectations such governance must satisfy.

Actors Governance leaders, compliance and risk functions, architects, delivery teams, operational managers, and responsible-AI practitioners involved in shaping, implementing, and overseeing delegation decisions.

Transformation From implicit, fragmented, and reactive delegation of system authority to explicit, structured, and reviewable governance of delegated authority over time.

Weltanschauung As A-AI systems move from advisory roles towards workflow-embedded execution, governance must address not only whether a system is acceptable to deploy, but what it is permitted to do, under which conditions, and how that permission should be revised as the system evolves.

Owners Enterprise governance and leadership functions with formal authority over risk, compliance, technology, and strategic deployment choices.

Environmental constraints Regulation, sectoral obligations, delivery pressure, uneven organisational maturity, distributed ownership structures, and uncertainty about acceptable levels of autonomy.

The CATWOE analysis clarified that the core transformation was neither purely technical nor purely regulatory. The governance problem was one of coordinated organisational control over delegated authority. This insight helped distinguish the study from a generic responsible-AI framework or a technical agent design exercise. It also clarified why the problem could not be resolved through a single policy or review instrument: the transformation required explicit linkage between strategic intent, operational permissions, monitoring, and revision.

The resulting root definition was formulated as follows:

> *A governance system, operated by enterprise organisations deploying A-AI under regulatory and risk constraints, that explicitly decides what operational authority may be delegated to adaptive systems, specifies and constrains that authority in a reviewable form, monitors whether the system remains within its authorised scope, and revises delegation when operational evidence indicates that scope has changed, so that value can be pursued without losing accountability, compliance, or control over evolving system behaviour.*

This root definition provided the normative anchor for the subsequent problem analysis. It did not prescribe the final architecture directly, but it established the minimum governance ambition against which both the literature and the interview material could be assessed. In particular, it made clear that the problem concerned more than initial approval: it concerned the full lifecycle of delegated authority, including specification, monitoring, and revision. The governance gaps derived from this diagnosis are presented in Chapter 2, and the architectural response is developed in Chapter 4.

Appendix C

DGA Design

This appendix presents the requirements traceability table and the supporting interaction and functional decomposition diagrams for the DGA.

C.1 Traceability Table

Table C.1: Function-level requirements traceability for the DGA.

Function ID	Function name	Gap(s)	Design requirement	Primary output
F0	Specify and maintain delegated authority	G1–G5	The architecture must specify, govern, and maintain delegated authority across the lifecycle of an agentic AI deployment.	Updated governance specifications and versioned DDA
F1	Calibrate authority lifecycle	G4, G5	The architecture must support lifecycle review of delegated authority when governance-relevant change, evidence, or review requirements indicate that the current governance state may no longer be adequate.	Authority review decision; versioned DDA update

Table C.1: Function-level requirements traceability for the DGA (continued)

Function ID	Function name	Gap(s)	Design requirement	Primary output
F11	Assess authority alignment	G4, G5	The architecture must assess whether observed behaviour, usage, or contextual change remains aligned with the authority currently in force.	Authority alignment assessment
F111	Assess deviation significance	G4	The architecture must determine whether identified deviations represent expected variation or governance-relevant change.	Deviation significance assessment
F112	Map evidence to current governance state	G4	The architecture must relate incoming governance evidence to the current governance baseline in order to identify relevant deviations.	Governance state deviation map
F113	Classify authority impact	G4, G5	The architecture must classify the likely impact of observed change on the current delegated authority.	Evidence-based change classification
F12	Determine revision pathway	G4, G5	The architecture must determine whether the current authority arrangement remains valid, requires refinement, or requires broader re-authorisation.	Authority review decision
F13	Ingest and triage governance triggers	G4	The architecture must assess incoming governance triggers and evidence in order to determine whether authority review is required and whether the available evidence is sufficient.	Triaged governance trigger basis; operational evidence retrieval request
F14	Issue versioned DDA update	G5	The architecture must record lifecycle review outcomes and resulting governance changes in a versioned governance record.	Versioned DDA update
F2	Specify authority boundary	G1, G2, G4	The architecture must define the outer limits within which delegated authority may be considered for a given deployment context.	Authority Boundary Specification (update)
F21	Develop authority boundary constraints	G1, G2	The architecture must define the conditions and limits under which delegation remains governable.	Authority boundary constraints

Table C.1: Function-level requirements traceability for the DGA (continued)

Function ID	Function name	Gap(s)	Design requirement	Primary output
F22	Translate deployment context into delegation scope	G1	The architecture must translate deployment context into an explicit delegation scope.	Delegation scope
F23	Establish prohibited authority register	G2	The architecture must explicitly identify action classes that must not be delegated.	Prohibited authority register
F24	Define change-significance criteria	G4	The architecture must define the conditions under which observed change should trigger governance review.	Change-significance criteria
F3	Determine delegated authority specification	G1, G2, G3	The architecture must specify what authority is actually delegated within the defined boundary.	Delegated Authority Specification (update)
F31	Specify escalation routing and delegation conditions	G3	The architecture must define the conditions under which actions may proceed autonomously and how escalation must occur where required.	Escalation routing and delegation conditions
F32	Determine viable delegation space	G1, G2	The architecture must identify the subset of the bounded action space that is viable for delegation.	Viable delegation space
F33	Map system capabilities to authority boundary	G1, G2	The architecture must assess how the system capability profile aligns with the defined authority boundary.	Capability-boundary alignment map
F34	Classify delegated actions	G2, G3	The architecture must classify viable actions into governance-relevant categories such as permitted, escalation-required, and prohibited.	Delegated action classification
F4	Define monitoring and enforcement configuration	G3, G4	The architecture must specify how delegated authority is monitored, interpreted, and governed in operation.	Monitoring and Enforcement Configuration (update)

Table C.1: Function-level requirements traceability for the DGA (continued)

Function ID	**Function name**	**Gap(s)**	**Design requirement**	**Primary output**
F41	Specify monitoring scope and signals	G3, G4	The architecture must define what actions, states, and interactions must be observable to govern delegated authority in practice.	Monitoring requirements set
F42	Define signal interpretation and trigger logic	G3, G4	The architecture must define how observed signals are interpreted and when they trigger escalation, containment, or review.	Governance trigger logic
F43	Specify enforcement responses and escalation handling	G3	The architecture must define what governance response follows when monitored conditions indicate that action is required.	Enforcement response configuration
F44	Assess monitoring and enforcement feasibility	G3, G4	The architecture must assess whether the proposed monitoring and enforcement arrangement can be realised consistently in practice.	Monitoring and enforcement feasibility assessment

C.2 Interaction Diagrams

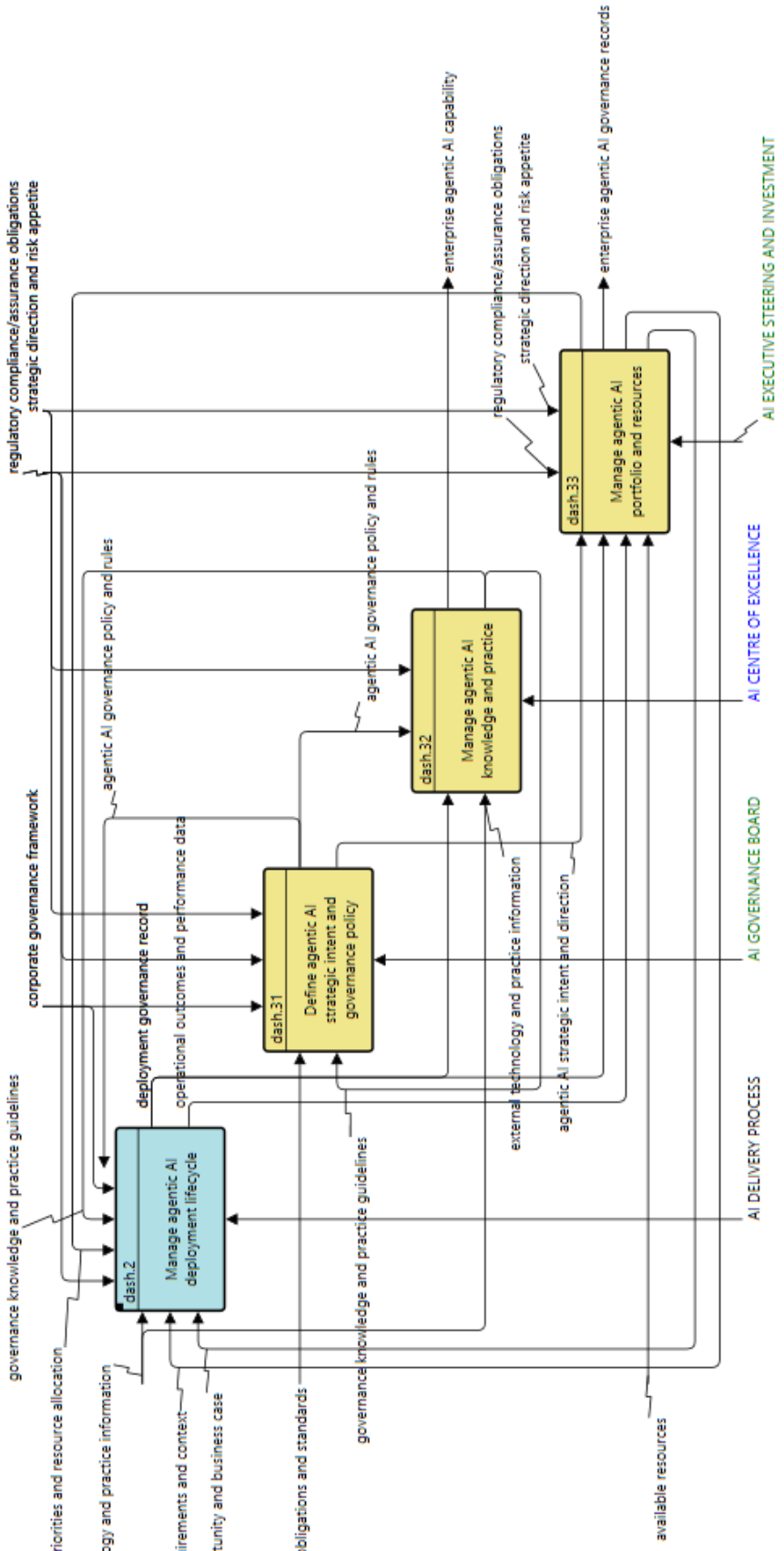

Figure C.1: Enterprise governance functions surrounding the DGA (highest-level context). The system-of-interest is shown in blue; surrounding governance functions in yellow. Labels along the bottom identify the organisational roles or governance bodies responsible for each function. Inputs enter from the left, controls from above, and outputs exit to the right.

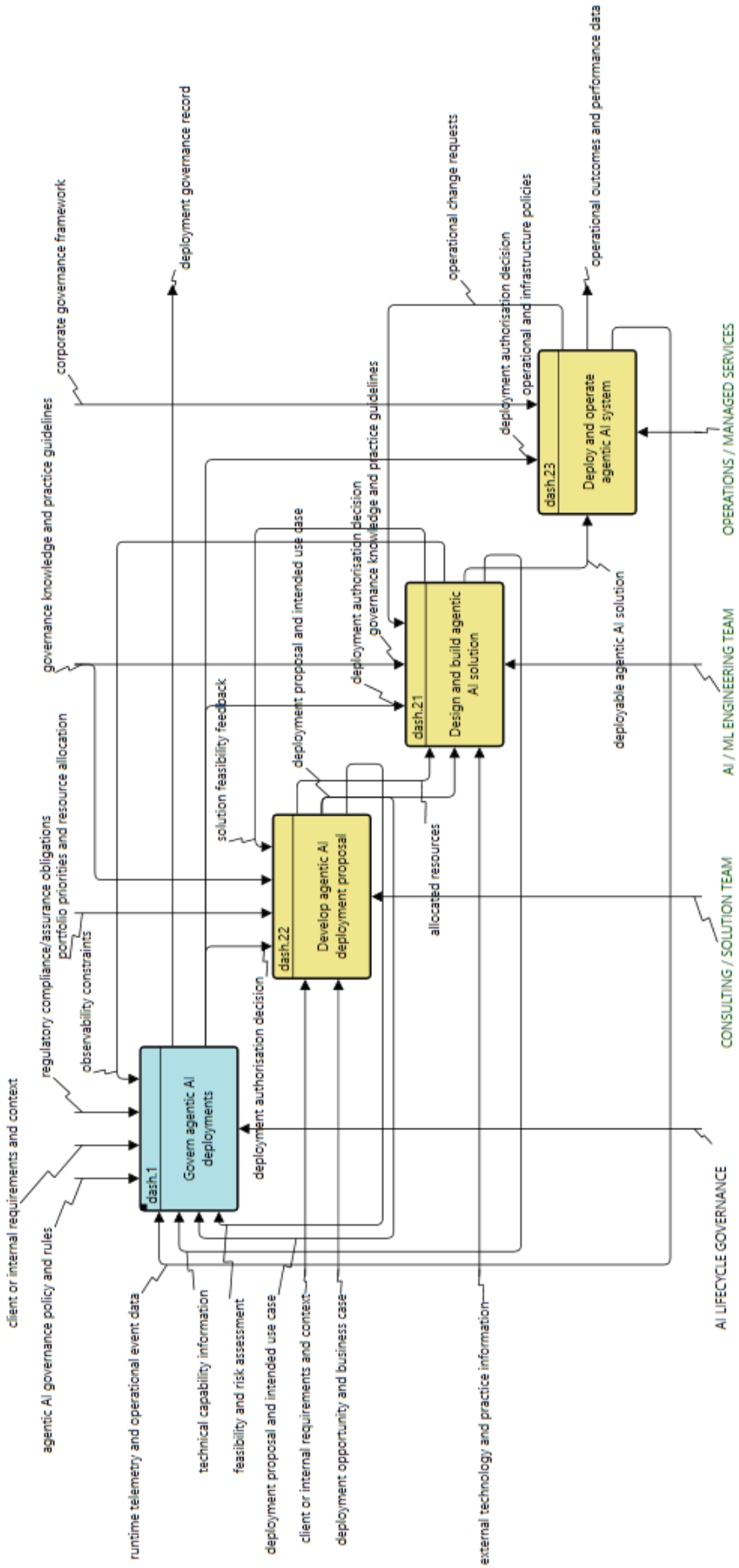

Figure C.2: Deployment lifecycle functions surrounding the DGA (middle-level context). The system-of-interest is shown in blue; surrounding governance functions in yellow. Labels along the bottom identify the organisational roles or teams responsible for each function. Inputs enter from the left, controls from above, and outputs exit to the right.

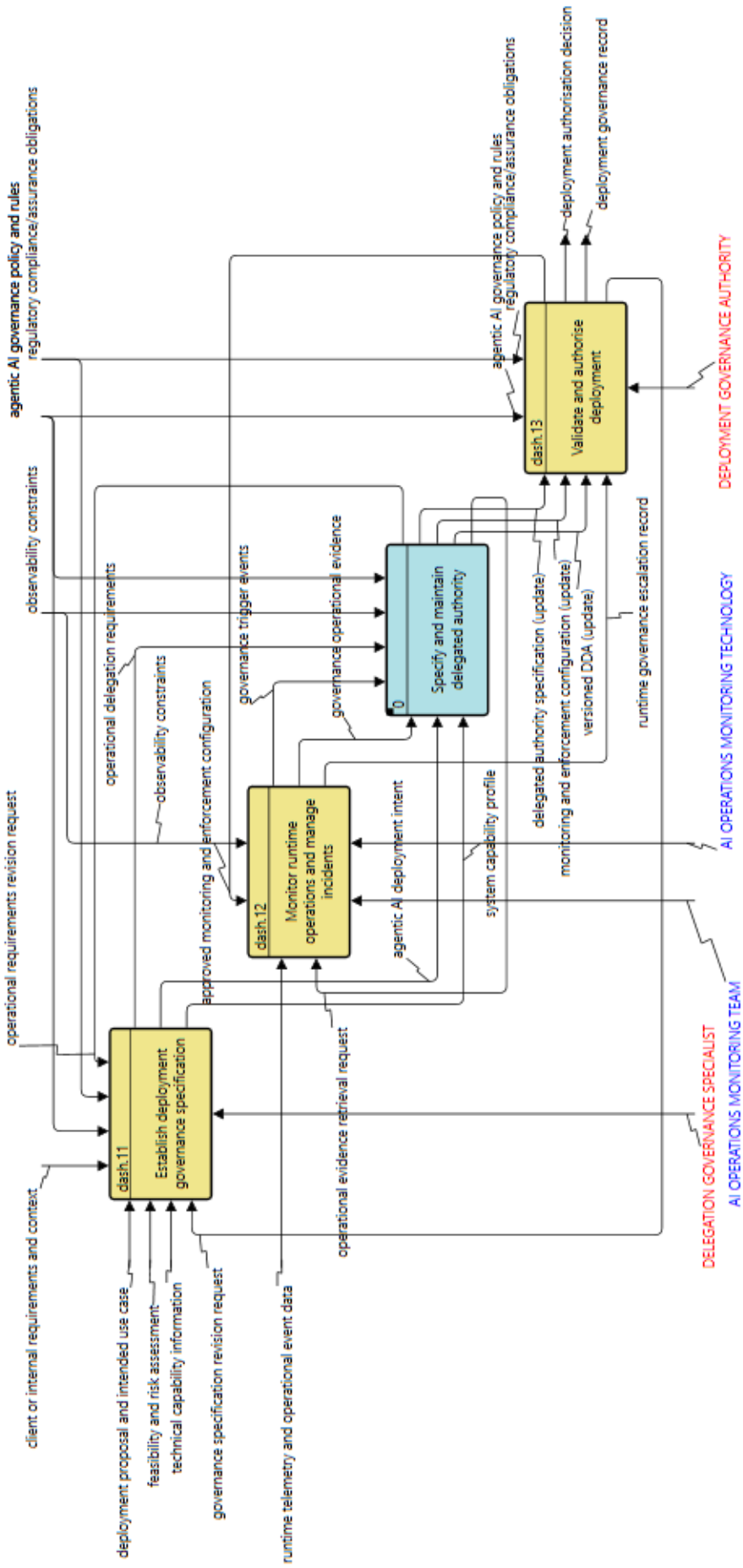

Figure C.3: Deployment governance functions surrounding the DGA (lowest-level context). The system-of-interest is shown in blue; surrounding governance functions in yellow. Labels along the bottom identify the organisational roles or technology responsible for each function. Inputs enter from the left, controls from above, and outputs exit to the right.

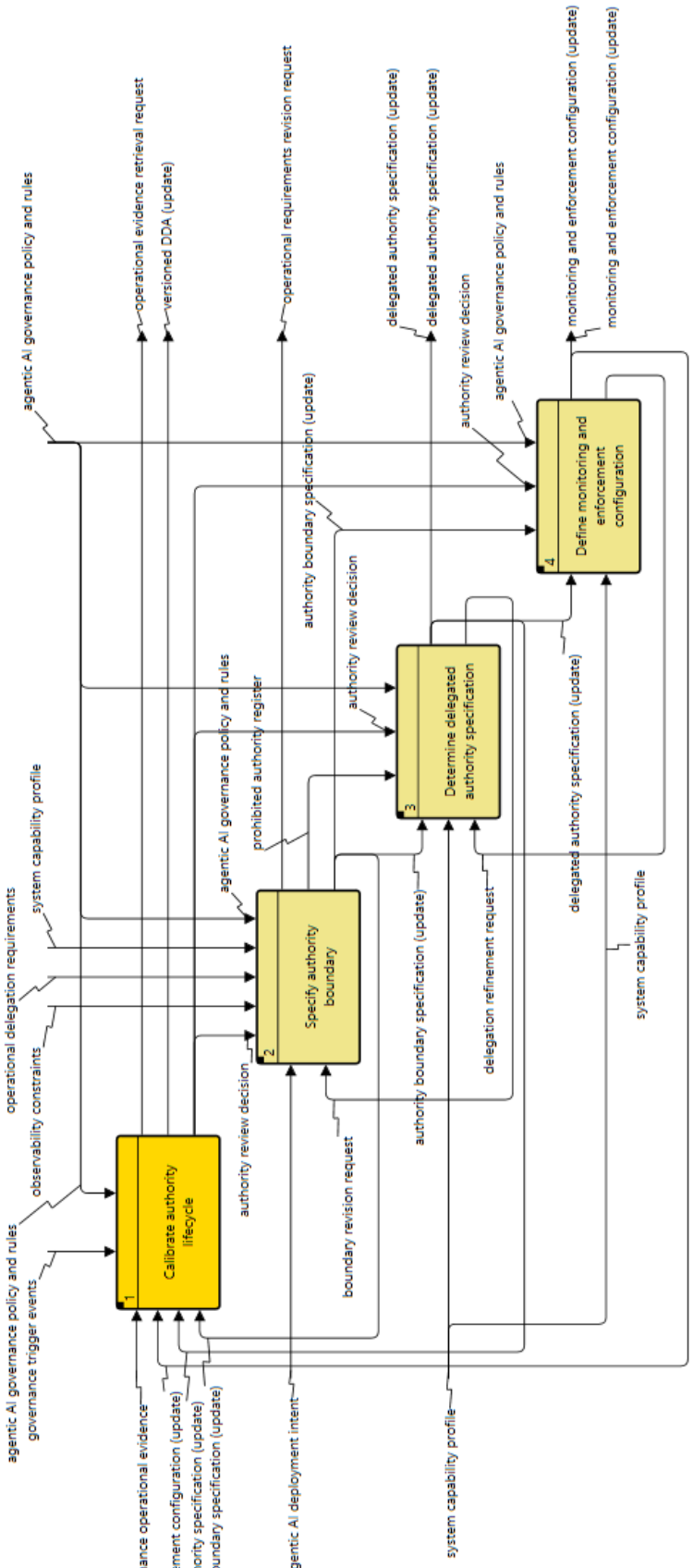

Figure C.4: First-level functional decomposition of the DGA. The lifecycle calibration function (F1, darker shading) operates across the three specification functions (F2–F4, lighter shading), ingesting governance trigger events and operational evidence, determining whether review is required, and issuing authority review decisions or targeted revision requests. Together, the functions update the versioned DDA and may request additional operational evidence where the existing evidence base is insufficient.

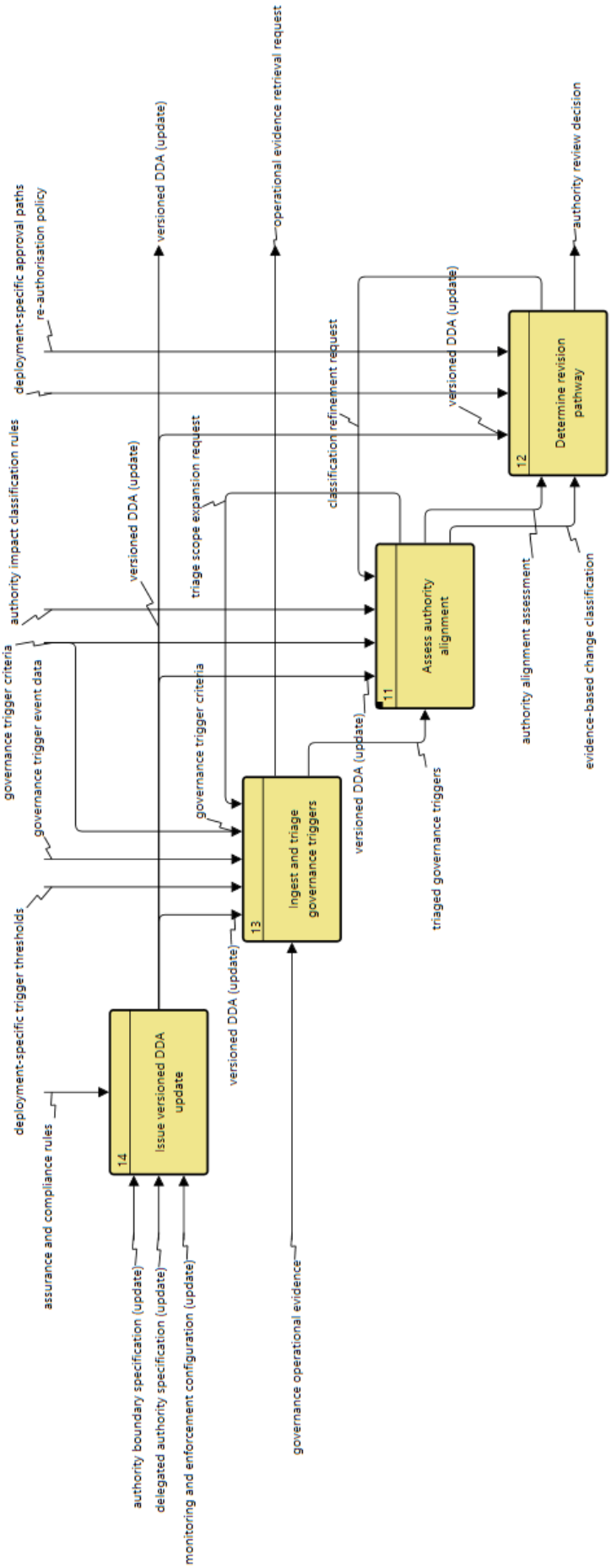

Figure C.5: Functional decomposition of "Calibrate authority lifecycle" (F1). Four sub-functions ingest and triage governance triggers, assess authority alignment, determine an authority review decision, and issue a versioned update of the DDA.

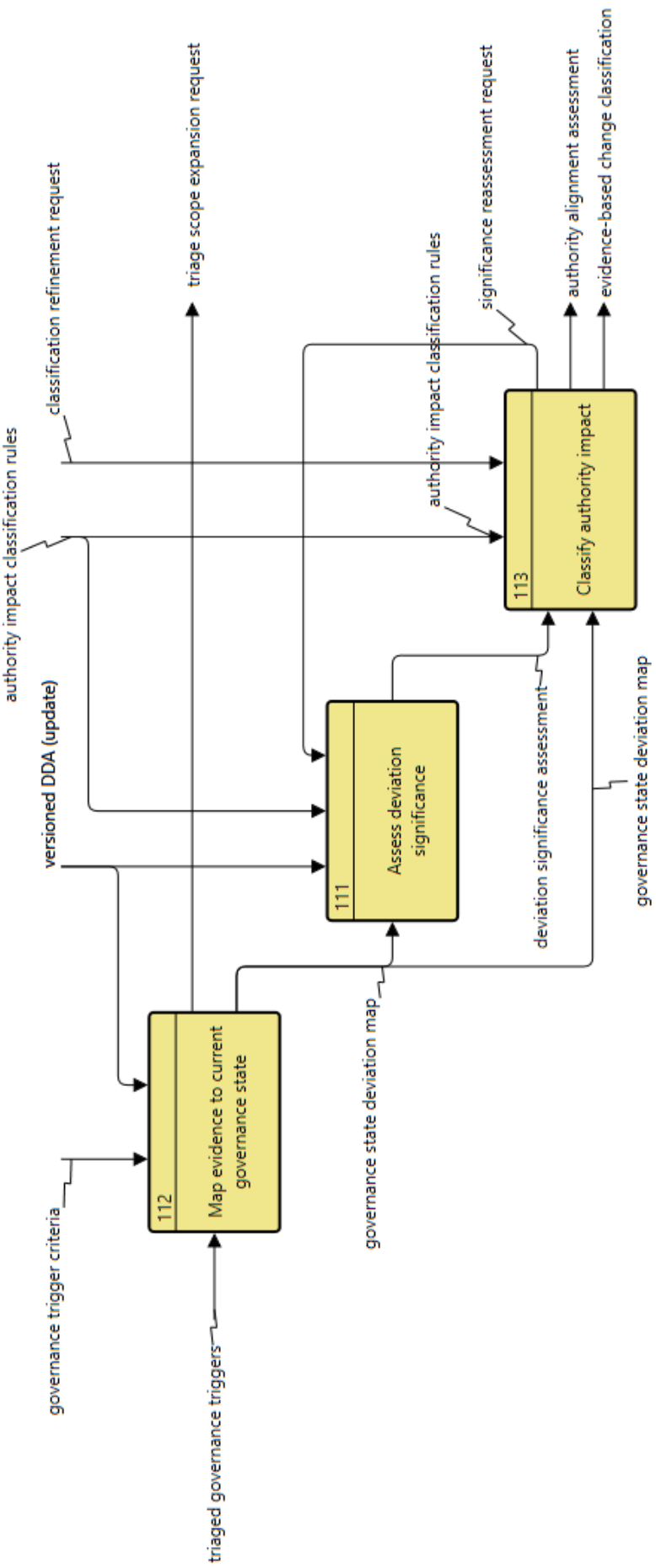

Figure C.6: Functional decomposition of "Assess authority alignment" (F11). Three sub-functions map incoming evidence to the current governance state, assess the significance of observed deviations, and classify their impact on delegated authority.

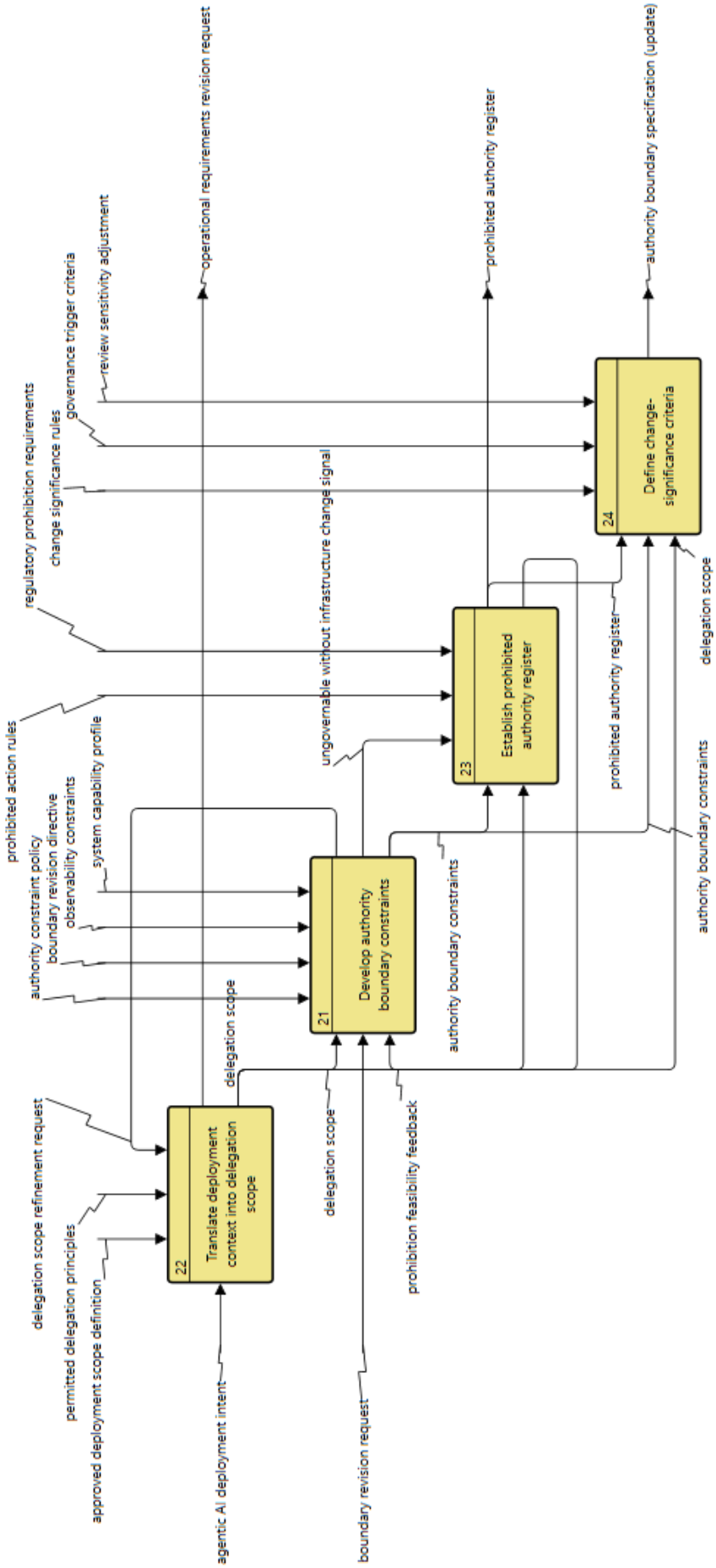

Figure C.7: Functional decomposition of "Specify authority boundary" (F2). Four sub-functions translate deployment intent into a delegation scope, develop boundary constraints, establish the prohibited authority register, and define change-significance criteria.

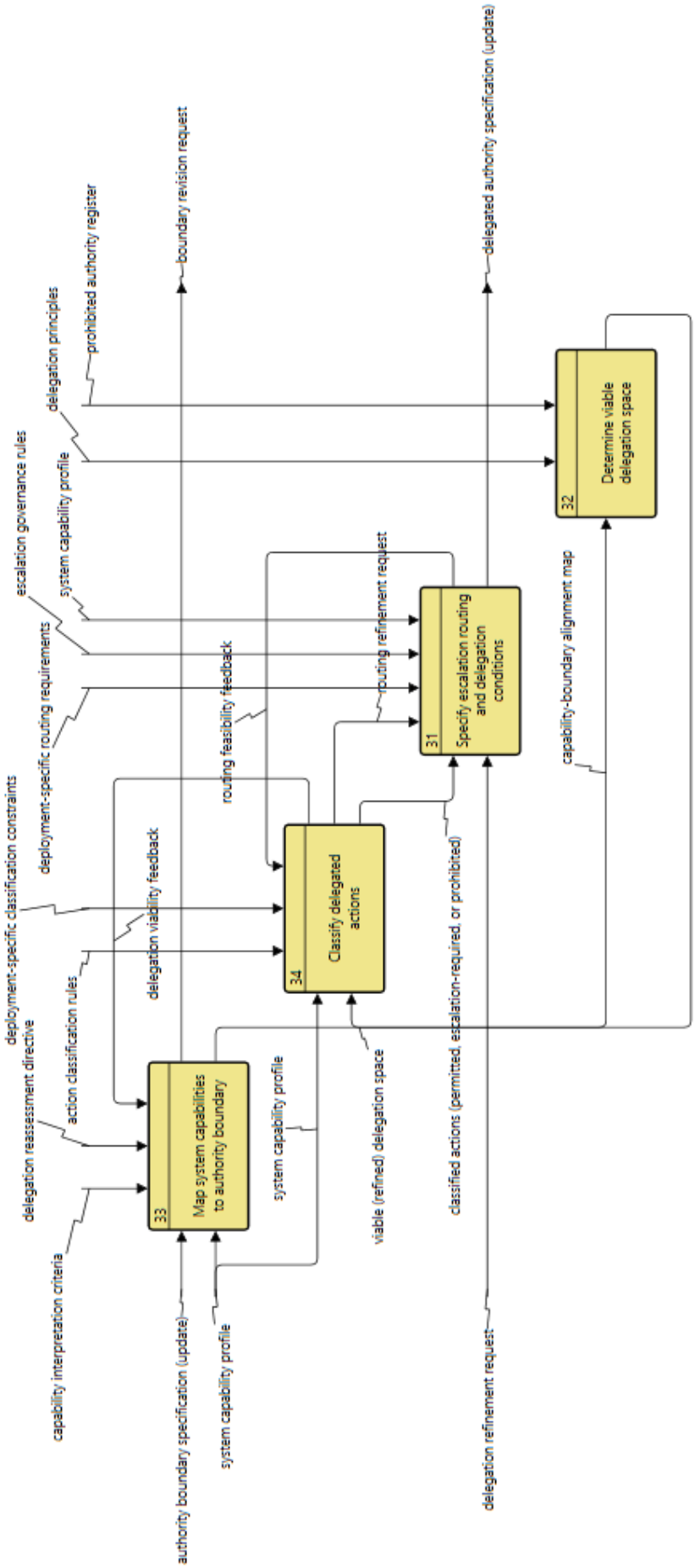

Figure C.8: Functional decomposition of "Determine delegated authority specification" (F3). Four sub-functions map system capabilities to the authority boundary, determine the viable delegation space, classify delegated actions, and specify escalation routing and delegation conditions.

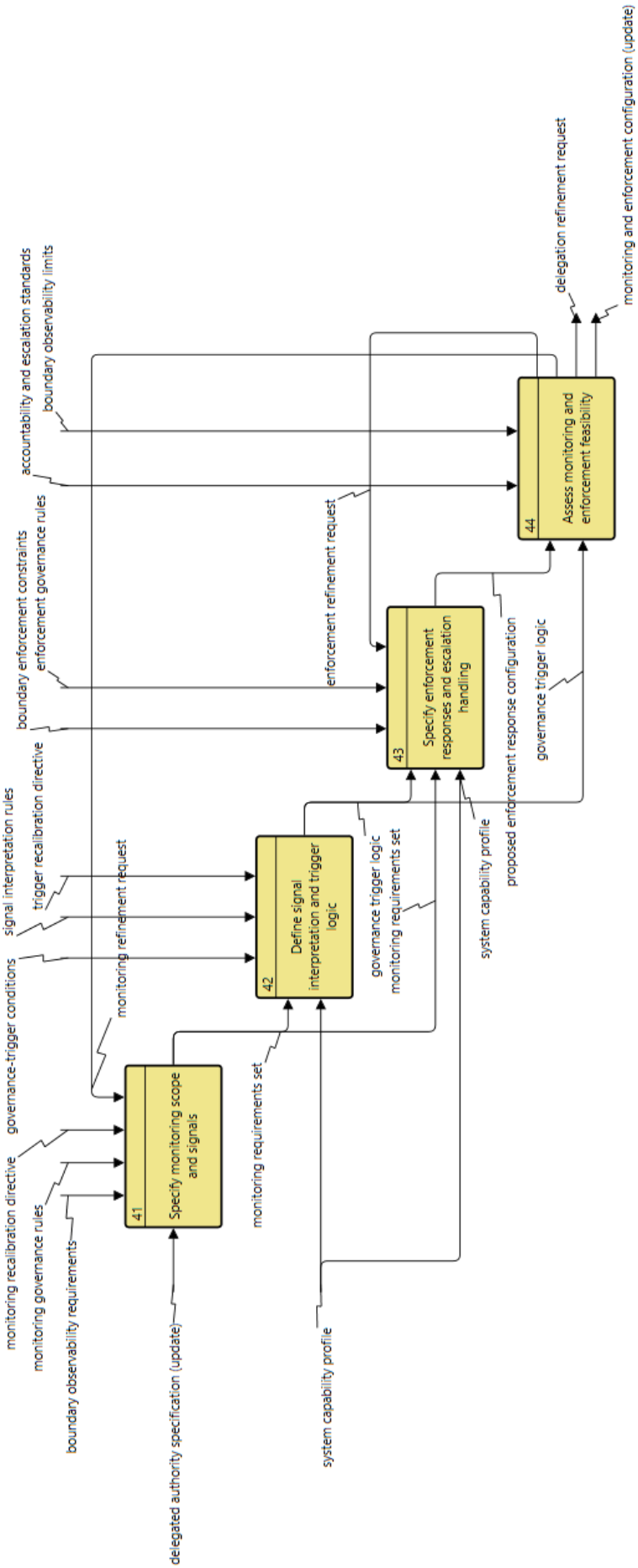

Figure C.9: Functional decomposition of "Define monitoring and enforcement configuration" (F4). Four sub-functions specify monitoring scope and signals, define signal interpretation and trigger logic, specify enforcement responses and escalation handling, and assess the feasibility of the resulting monitoring and enforcement configuration.