ADDENDUM B to the Chief Enterprise Architect Practitioner's Guide

Expanding Section 3 of the CEA Practitioner's Guide

Covers all enterprise-level, domain-level, and project-level deliverables:

purpose • structure • content requirements • quality standards

CEA role • who produces it • stakeholders • tools • common pitfalls

Digital Enterprise Architecture Advisors LLC

Enterprise Architecture Leadership

Version: 1.0
Date: July 2026
Author: Christian Kobsa; Digital Enterprise Architecture Advisors LLC

Abstract

Addendum B expands Section 3 of the *CEA Practitioner's Guide* by providing a concise, practitioner-ready reference for all enterprise-, domain-, and project-level architecture deliverables. It explains each artifact's purpose, required structure, ownership, quality standards, and common pitfalls, grounding every deliverable in business-aligned architectural practice.

At the enterprise tier, it details the seven core artifacts that define architectural direction — including the Architecture Vision ("a board member with no technology background can read the executive summary and explain the direction"), Architecture Principles, the Business Capability Model, and the Application Portfolio. Domain-level sections translate enterprise intent into domain guidance, while project-level sections ensure architectural integrity during solution delivery.

Across all tiers, Addendum B reinforces that architecture artifacts are governance instruments, not documents: their value lies in clarity, traceability to strategy, and disciplined use in decision-making.

Table of Contents

Introduction to This Addendum

Section 3 of the CEA Practitioner's Guide introduced the full catalog of architecture deliverables across three tiers: enterprise-level, domain-level, and project-level. This addendum provides the deep-dive reference that Section 3 set out to introduce — treating each deliverable as a practitioner's topic in its own right.

For each deliverable, this addendum covers: its strategic purpose and how it creates value, the CEA's specific role in producing or overseeing it, the detailed structure and content requirements, the stakeholders who consume it and what they need from it, the tools used to produce and maintain it, quality standards and maturity indicators, and the most common pitfalls that reduce its effectiveness.

The deliverables are organized into three tiers mirroring the main guide:

- Section B.1 — Enterprise-Level Deliverables: the seven core artifacts the central EA team produces to define and govern the enterprise architecture
- Section B.2 — Domain-Level Deliverables: the six artifact types produced by domain architects within their specific business or technology domains
- Section B.3 — Project-Level Deliverables: the six artifact types produced by solution architects on specific delivery programs

> **How This Addendum Relates to Addendum A**
>
> Addendum A describes what the CEA does in each TOGAF ADM phase. This addendum describes the artifacts those phases produce. The two documents are complementary: Addendum A answers 'when and why' deliverables are created; Addendum B answers 'what they contain and how to make them excellent.'

Deliverables At a Glance

Tier	Deliverable	Primary Owner	ADM Phase(s)	Update Frequency
Enterprise	Architecture Vision & Strategy	CEA	Phase A	Each ADM cycle / major strategy refresh
Enterprise	Architecture Principles	CEA	Preliminary / Phase A	Annual review; update as strategy evolves

Tier	Deliverable	Primary Owner	ADM Phase(s)	Update Frequency
Enterprise	Business Capability Model	Business Architecture Lead	Phase B	Annual review; update on org/strategy change
Enterprise	Application Portfolio	Application Architecture Lead	Phase C	Continuous; formal review bi-annually
Enterprise	Technology Radar & Standards Register	Infrastructure & Cloud Lead	Phase D	Bi-annual publication
Enterprise	Architecture Roadmap	CEA	Phases E & F	Quarterly alignment; major refresh annually
Enterprise	Architecture Decision Records (ADRs)	Domain / Solution Architects	All phases	Created per significant decision; permanent record
Domain	Domain Capability Maps	Business Domain Architects	Phase B	Annual; updated on domain strategy change
Domain	Domain Application Landscapes	Platform / Domain Architects	Phase C	Bi-annual; updated on portfolio change
Domain	Domain Data Models	Data Architecture Lead + Domain	Phase C	Updated on data architecture change
Domain	Domain Integration Architectures	Application Architecture Lead	Phase C	Updated on integration pattern change
Domain	Regional Architecture Variations	Regional Architects	All phases	Updated on regulatory or strategy change
Domain	Domain Roadmaps	Domain Architects	Phases E & F	Quarterly alignment with enterprise roadmap

Tier	Deliverable	Primary Owner	ADM Phase(s)	Update Frequency
Project	Solution Architecture Document (SAD)	Solution Architects	Phase G	Per project; updated at major design changes
Project	Architecture Compliance Assessment	Domain / Solution Architects	Phase G	Per review gate; typically 2–3 per project
Project	High-Level Design (HLD)	Solution Architects	Phase G	Per project; updated at significant changes
Project	Low-Level Design (LLD)	Engineering Leads	Phase G	Per component; updated during build
Project	Non-Functional Requirements Spec	Solution Architects	Phase G	Per project; baselined at design sign-off
Project	Integration Design Document	Solution / Integration Architects	Phase G	Per integration; updated on interface change

Section B.1

Enterprise-Level Deliverables

The seven core artifacts produced by the central EA team to define and govern the enterprise architecture

Enterprise-level deliverables are the defining outputs of the CEA and the central EA team. They establish the architectural direction, standards, and investment priorities that govern the entire organization. Each one is a living document — not a one-time publication — and the CEA is personally accountable for their quality, currency, and business relevance.

B.1.1 Architecture Vision and Strategy

Purpose and Strategic Value

The Architecture Vision is the CEA's primary strategic communication instrument. Its purpose is to create and sustain executive alignment around a coherent direction for the organization's technology landscape. Without a clear, compelling Architecture Vision, the EA practice operates in a vacuum — producing technically sound artifacts that nobody has committed to acting on.

The Architecture Vision serves three distinct audiences simultaneously:

- C-suite and board — it presents the technology direction in business outcome language, making the case for investment and change
- Business unit leaders — it shows how the enterprise architecture will improve their specific capability performance and eliminate pain points
- Technology teams — it provides a north star that guides hundreds of individual technical decisions made daily across the organization

> **The CEA's Test for a Good Architecture Vision**
>
> A well-crafted Architecture Vision passes three tests: (1) A board member with no technology background can read the executive summary and explain the direction to a colleague. (2) A project team lead can consult it to determine whether a proposed technical approach is consistent with the strategic direction. (3) The CFO can use it to frame an investment case. If the Vision fails any of these tests, it needs rework.

CEA's Role

The Architecture Vision is one of the few deliverables the CEA must personally author in substance — not just review. The strategic synthesis it requires — translating business intent into architectural direction — cannot be meaningfully delegated. The CEA will typically work closely with the Business Architecture Lead and Deputy EA to develop the domain-specific content, but the overall narrative, the priorities, and the framing are the CEA's personal expression of architectural judgment.

Detailed Structure and Content Requirements

Section	Content	Length / Format
Executive Summary	Business context in 3–5 sentences; the architectural imperative (why change is necessary now); the target state in one	1 page maximum; plain business language

Section	Content	Length / Format
	paragraph; the top 3–5 investment priorities. Written for a non-technical executive audience.	
Business Context and Strategic Drivers	Description of the organization's business model and value chain; the strategic priorities driving the architecture engagement (growth, efficiency, compliance, digital transformation); the architectural implications of each driver.	2–3 pages; structured around business themes, not technology
Architecture Goals and Success Metrics	5–10 specific, measurable architectural outcomes tied to business value. Each goal should have a named metric and a target state. Example: 'Reduce application duplication from 340 to under 200 systems within 3 years, reducing annual support cost by £12M.'	Half to one page; table format recommended
Current Architecture Assessment	Honest assessment of the current architecture's strengths, weaknesses, and risks. Key pain points, technical debt hot spots, and systemic vulnerabilities. Do not sugarcoat — executive credibility depends on honesty here.	1–2 pages; heat map or RAG status visual recommended
High-Level Target Architecture	Conceptual views of the future architecture across all four domains (business, data, application, technology). These are not detailed designs — they are directional pictures showing the major components and their relationships.	2–4 pages; primarily visual with explanatory captions
Key Architectural Decisions	The major architectural choices already made and the rationale behind them. Cloud platform strategy, platform consolidation decisions, integration architecture style, data platform direction.	1–2 pages; decision table with rationale and alternatives considered
Roadmap Summary	A one-page visual showing the major waves of change, their timing, and the business capability each wave enables. This is the bridge between the vision and the investment plan.	1 page; timeline visual with business value callouts
Risk Summary	The 5–10 most significant architectural risks — both the risk of acting (transformation	1 page; risk register table

Section	Content	Length / Format
	risk) and the risk of not acting (status quo risk). Each risk should have an owner and a stated mitigation approach.	
Governance and Engagement Model	How the architecture will be governed, how stakeholders will be kept informed, and how the CEA's team will support delivery teams in implementing the vision.	Half page; brief and practical

Quality Standards

- Business language throughout — no unexplained acronyms, no technology jargon in executive sections
- Every architectural goal traceable to a named business driver or strategic priority
- The roadmap summary consistent with the investment envelope agreed with the CFO
- Reviewed and signed off by at least the CIO and one senior business leader before publication
- Updated within 90 days of any material change to business strategy or architectural direction

Stakeholders and Their Needs

Stakeholder	What They Need from the Architecture Vision
CEO / COO	Confidence that the technology direction enables the business strategy; clarity on the investment required and the business outcomes it delivers
CFO	A credible investment case; clarity on what is capital vs. operating expenditure; the cost of inaction
CIO / CTO	Alignment between the architecture direction and their delivery and operations responsibilities; clarity on what they are being asked to commit to
Business Unit Leaders	Evidence that their domain's pain points are understood and addressed; clarity on what will change in their domain and when
Program / Delivery Leadership	A clear direction that guides program scoping and technical decision-making; enough specificity to avoid architectural dead ends
Board / Risk Committee	Assurance that the technology landscape is under strategic control; visibility of the major risks being managed

> ⚠ **Common Pitfall: The Technology Showcase**
>
> The most common failure mode for Architecture Visions is that they become showcases for technology rather than expressions of business intent. A vision dominated by cloud platform diagrams, microservices architecture patterns, and DevOps toolchain descriptions will fail to engage business stakeholders and will be dismissed as 'an IT thing.' The CEA must ensure that business capability language dominates, and that every technology choice is framed in terms of the business problem it solves.

B.1.2 Architecture Principles

Purpose and Strategic Value

Architecture Principles are the constitutional layer of the EA practice. They are the foundational commitments the organization makes about how technology decisions will be made — commitments that persist across individual projects, budget cycles, and even strategic pivots. Good principles create consistency without rigidity; they guide decisions rather than dictating them.

Principles serve three functions in the architecture practice: they provide decision-making guidance for architects and delivery teams facing choices where no specific standard exists; they provide the rationale for governance decisions — every ARB ruling should be traceable to one or more principles; and they provide the language for communicating architectural values to non-technical stakeholders.

CEA's Role

The CEA is the author and ultimate owner of the Architecture Principles. While their development should be collaborative — engaging business leaders, technology peers, and the architecture team — the CEA must personally vouch for each principle and be prepared to defend it in executive forums. Principles that the CEA cannot defend with conviction will not survive first contact with a determined project sponsor seeking an exception.

Detailed Structure: The Five-Component Principle

Every principle must have all five components fully developed. A principle with a name and a statement but no implications is incomplete and will be ineffective.

Component	Guidance for Writing	Quality Test
Name	2–4 words; memorable; captures the essence of the principle. Avoid jargon. Examples: 'Cloud First', 'Reuse Before Buy Before Build', 'Data as an Asset', 'Design for Failure.'	Can a non-architect recite the name from memory after hearing it once?
Statement	One clear, unambiguous sentence. Active voice. Avoid 'should' — use 'will' or 'must' to signal genuine commitment. The statement must be actionable: it should be possible to assess whether a given decision is consistent with it.	Can an architect use this statement to rule a specific proposal in or out?

Component	Guidance for Writing	Quality Test
Rationale	2–5 sentences explaining the business value the principle creates or the risk it mitigates. Written for a business audience. Should reference a strategic driver where possible.	Would a CFO or business unit leader agree this is a good reason to commit to this principle?
Implications	A bulleted list of the concrete changes in behavior, process, or investment that the principle requires. This is the most important component — and the most often skipped. Implications must be honest about what the principle costs, not just what it delivers.	Does at least one implication describe something the organization currently does NOT do, but will need to do?
Metrics	1–3 measurable indicators that show the principle is being followed. Tied to existing data sources where possible.	Can the EA team produce a quarterly report on this metric from available data?

The Principles Register: Format and Governance

Principles should be maintained in a formal register — not in a slide deck or email thread. The register is a controlled document owned by the CEA, with formal version control and an annual review cycle. Key register governance requirements:

- Version numbering — every change to a principle is tracked with a version number, change date, and change rationale
- Status field — Active, Under Review, Superseded, or Retired; no principle is simply deleted
- Conflict register — where two principles create tension (e.g., 'Cloud First' vs. 'Data Sovereignty'), the conflict is explicitly documented with guidance on how to resolve it
- Exception log — when a project is granted an exception to a principle, it is logged against the principle with the business justification and remediation commitment

Recommended Principle Categories for a Large Multinational

Category	Example Principles	Typical Count
Business & Technology Alignment	Architecture decisions will be driven by business capability priorities. Technology investment will be tied to named business outcomes.	2–3

Category	Example Principles	Typical Count
Cloud & Infrastructure	Cloud-first for all new workloads. No new data center capacity will be commissioned without CTO approval.	2–4
Application & Integration	Reuse before buy before build. APIs are the default integration mechanism. No point-to-point integrations.	3–5
Data & Information	Data is an enterprise asset, not a departmental resource. Master data has a single authoritative source. Privacy by design.	3–5
Security & Compliance	Security is a first-class architecture concern addressed from inception, not retrofit. Least privilege access by default.	2–3
Governance & Standards	All technology decisions above a defined threshold require architecture review. Exceptions are granted, not ignored.	2–3

Testing Your Principles

Before ratifying a set of principles, the CEA should test them against 5–10 real recent technology decisions made in the organization. For each decision, ask: would this principle have changed the outcome? If the answer is 'no' for all test decisions, the principle adds no value. If the answer is 'yes' for many decisions, the principle is doing genuine work. A set of principles that would have prevented 3–4 of the last 10 poor technology decisions is a valuable governance tool.

B.1.3 Business Capability Model

Purpose and Strategic Value

The Business Capability Model (BCM) is the most strategically significant artifact in the EA toolkit. It provides a stable, business-language view of what the organization must be able to do — completely independent of organizational structure, processes, roles, or technology. Because capabilities are stable even when organizational structures and systems change, the BCM provides the fixed reference point around which all other architecture artifacts are anchored.

The BCM enables three classes of architecture decision that are otherwise extremely difficult to make rigorously:

- Investment prioritization — by assessing each capability's strategic importance and current performance, the CEA can produce a defensible, evidence-based case for where architecture investment should be concentrated
- Application rationalization — by mapping applications to the capabilities they support, portfolio decisions (invest, migrate, retire) can be made based on business value rather than technical affinity or historical momentum
- Transformation sequencing — by identifying which capabilities are most critical to strategic objectives and most under-served by current technology, the CEA can sequence transformation programs to maximize business value delivered early

CEA's Role

The CEA is the sponsor and quality owner of the BCM. The detailed construction is led by the Business Architecture Lead, but the CEA must personally validate the Level 1 capability domains with senior business leaders and ensure the heat map assessment reflects the organization's genuine strategic priorities. The CEA also owns the governance of the BCM — ensuring it is updated when strategy changes and that it is actively used in portfolio and investment decisions rather than sitting on a shelf.

Detailed Structure and Construction Approach

Level 1: Capability Domains

Level 1 defines the major domains of organizational activity — typically 8 to 15 for a large multinational. These should be named in business language, not IT language, and should represent genuine domains of business activity rather than organizational units.

Common Level 1 domains for a diversified multinational:

Domain Category	Examples
Customer-Facing	Customer Acquisition, Customer Management, Customer Service & Support
Product & Service	Product Development, Product Management, Service Delivery
Operations	Supply Chain Management, Manufacturing / Operations, Logistics & Distribution
Support Functions	Finance & Accounting, Human Capital Management, Legal & Compliance, Procurement
Enabling & Strategic	Technology & Innovation, Data & Analytics, Risk Management, Corporate Strategy

Level 2: Capability Groups

Level 2 decomposes each domain into 3–8 capability groups. The decomposition should follow natural business activity groupings — not mirroring the organizational chart. Example: Customer Management → Customer Acquisition, Customer Onboarding, Customer Retention, Customer Development, Customer Exit Management.

Level 3: Individual Capabilities

Level 3 defines the individual, atomic capabilities — specific things the organization must be able to do. Level 3 capabilities are the primary unit of analysis for application mapping and investment prioritization. Example: Customer Acquisition → Lead Generation, Lead Qualification, Opportunity Management, Proposal Management, Contract Negotiation.

Capability Heat Map Assessment

Each Level 3 (and sometimes Level 2) capability is assessed on two dimensions and plotted as a heat map:

Dimension	Assessment Criteria	Scale
Strategic Importance	How critical is this capability to the organization's competitive differentiation and strategic goals? Is it a commodity capability (table stakes) or a differentiating capability (competitive advantage)?	High / Medium / Low
Current Performance	How well does the organization currently perform this capability, relative to best-in-class competitors or internal expectations? Consider	Strong / Adequate / Weak

Dimension	Assessment Criteria	Scale
	process effectiveness, technology enablement, data quality, and talent.	

The combination of these two dimensions drives investment recommendation:

Strategic Importance	Current Performance	Investment Recommendation	Color
High	Weak	Invest urgently — close the gap	Red
High	Adequate	Invest to strengthen — competitive differentiator needs to be best-in-class	Amber
High	Strong	Maintain — protect what works	Green
Medium	Weak	Improve selectively — fix where it hurts most	Amber
Medium	Adequate / Strong	Tolerate / Automate — do not over-invest	Grey
Low	Any	Standardize / Outsource — commodity; minimize cost	Blue

Quality Standards and Maturity Indicators

- All Level 1 capabilities validated and named by senior business leaders — not just the EA team
- Heat map assessment corroborated by data (customer satisfaction scores, process performance metrics, system health scores) — not just subjective opinion
- Every significant application in the portfolio mapped to at least one Level 3 capability
- BCM reviewed and refreshed within 90 days of any major strategic change
- BCM actively referenced in portfolio review and investment committee discussions — if it is not being used in decisions, it is not fulfilling its purpose

> ⚠ **The Organizational Mirror Anti-Pattern**
> The most common BCM failure is building a capability model that mirrors the organizational chart — creating 'Finance capabilities,' 'HR capabilities,' and 'IT capabilities' rather than genuine business capabilities. This produces a model that is politically safe but architecturally useless, because it changes whenever the organization restructures. A good BCM describes what the organization does, not who does it.

B.1.4 Application Portfolio

Purpose and Strategic Value

The Application Portfolio is the CEA's primary tool for managing the organization's application landscape as a strategic asset. Most large multinationals have accumulated application estates of considerable complexity — hundreds or thousands of systems, many overlapping in function, many aging and costly to maintain, and many no longer aligned to the business capabilities they were designed to support.

An actively managed Application Portfolio enables the CEA to: quantify the total cost of ownership of the application estate; identify rationalization opportunities with clear business cases; prevent the further growth of portfolio complexity by enforcing 'one in, one out' or 'consolidate before adding' policies; and align application investment to strategic capability priorities.

CEA's Role

The CEA owns the Application Portfolio as a governance instrument and is the decision authority for portfolio classification and rationalization decisions. The detailed inventory management and assessment work is led by the Application Architecture Lead, but portfolio strategy decisions — particularly retire, consolidate, and replace decisions affecting business units — require CEA involvement to manage the organizational dynamics.

Building the Application Portfolio: Step by Step

1. Application Discovery — identify every application in use across the enterprise, including shadow IT. Sources include: CMDB, IT financial systems, software license inventories, network traffic analysis, and business unit interviews. In a large multinational, this discovery phase alone can take 2–4 months.
2. Data Collection — for each application, collect: application name, description, business owner, IT owner, primary business domain, supported capabilities (mapped to BCM), number of users, annual total cost of ownership (license, infrastructure, support), integration count, vendor, version, and end-of-support date.
3. Strategic Assessment — assess each application's business value: strategic alignment score (how well does it support differentiating capabilities?), business criticality (what is the impact of a 4-hour, 24-hour, 1-week outage?), user satisfaction score (NPS or equivalent), and regulatory importance (does it process regulated data or support compliance obligations?).
4. Technical Health Assessment — assess each application's technical condition: architecture fit (does it conform to enterprise standards?), technical debt level (estimated remediation cost as a multiple of annual run cost), vendor support status

(current / extended / end-of-life), integration complexity (number and criticality of integrations), cloud readiness (native cloud, cloud-compatible, cloud-resistant), and security posture.

5. Portfolio Classification — assign each application a portfolio strategy. The TIME model is the most widely used: Tolerate (low business value, acceptable technical health — run it cheaply), Invest (high business value, good technical health — grow and improve it), Migrate (high business value, poor technical health — modernize or replace), Eliminate (low business value — retire it).
6. Roadmap Integration — rationalization decisions from the portfolio feed directly into Phase E (Opportunities and Solutions) and the Architecture Roadmap. Each 'Migrate' or 'Eliminate' decision becomes a roadmap work package with a named business case.

Portfolio Visualization Models

Model	Description	Best Used For
TIME Matrix	2×2 matrix: Business Value (x-axis) vs. Technical Health (y-axis). Each application plotted as a bubble, sized by annual TCO. TIME quadrant labels applied. Color-coded by domain or classification.	Board and executive communication; portfolio prioritization discussion
Capability Coverage Map	BCM heat map overlaid with application counts per capability. Shows where the portfolio is over-served (too many apps per capability) and under-served (gaps).	Identifying rationalization opportunities; investment gap analysis
Application Lifecycle Timeline	Applications plotted on a timeline showing vendor end-of-support dates, contract renewal dates, and planned retirement or replacement dates.	Portfolio risk management; contract negotiation planning
Domain Application Landscape	Applications grouped by business domain, showing integrations between applications. Shows cross-domain dependencies and integration complexity.	Domain architecture planning; integration simplification initiatives
TCO Waterfall	Applications ranked by annual total cost of ownership. Shows the cumulative cost profile of the portfolio and identifies the highest-cost rationalization targets.	Financial case for portfolio rationalization; CFO communication

Portfolio Governance

An Application Portfolio that is built once and not maintained rapidly becomes an inaccurate liability rather than a useful governance tool. The CEA must establish a portfolio governance process:

- Continuous update process — any new application requires ARB approval and immediate portfolio registration; any application retirement requires portfolio update within 30 days
- Bi-annual formal review — a structured review of portfolio health metrics, rationalization progress, and emerging risks
- Portfolio investment committee — a regular forum where the CEA presents portfolio health and rationalization business cases for CFO and CIO decision
- Portfolio KPIs — metrics tracked and reported quarterly: total application count, applications in each TIME category, applications past vendor end-of-support, portfolio rationalization savings realized

B.1.5 Technology Radar and Standards Register

Purpose and Strategic Value

The Technology Radar is the CEA's primary tool for governing technology choices across the enterprise. In large organizations without an active Technology Radar, technology proliferation is the default outcome — each project team selects the technology they are most familiar or excited about, resulting in a fragmented, expensive, and hard-to-support landscape. The Technology Radar establishes a shared vocabulary of approved, emerging, and deprecated technologies that guides consistent decision-making.

The ThoughtWorks Technology Radar — published bi-annually since 2010 — popularized the radar format. The CEA adapts this concept to the organization's specific context, technology strategy, and governance model.

CEA's Role

The CEA is the governance authority for the Technology Radar. Publication decisions — particularly moving a technology from Trial to Adopt, or from Hold to Retire — carry significant organizational implications and require CEA sign-off. The detailed assessment work is led by the Infrastructure and Cloud Architecture Lead, with domain architects and senior engineers contributing assessments.

Radar Structure and Ring Definitions

Ring	Definition	Entry Criteria	Governance Implication
Adopt	Technologies that are proven, strategic, and approved for broad use across the enterprise. The default choice for new work in applicable domains.	Production-proven in the organization or industry; supported by the EA team; aligned to vendor long-term roadmap; training and support available.	All new work in the applicable domain must use Adopt-ring technologies unless a waiver is granted by the ARB.
Trial	Technologies being evaluated with real workloads. Not yet broadly recommended but approved for use in defined pilot contexts.	Promising technology with credible vendor roadmap; EA team has assessed it; a pilot project or proof of concept is in progress or planned.	Use permitted in defined pilot contexts only. Pilot outcomes must be reported to the EA practice. Technology will be moved to Adopt or Hold based on pilot results.
Assess	Technologies on the EA team's watch list.	The EA team has become aware of the technology	No production use. EA team monitors vendor roadmap,

Ring	Definition	Entry Criteria	Governance Implication
	Interesting but not yet sufficiently proven or strategic to recommend for active use.	through industry analysis, vendor engagement, or community activity. No production use yet.	industry adoption, and community activity. Annual reassessment.
Hold	Technologies the organization is actively moving away from. No new investment; existing usage to be migrated on a defined timeline.	Technology is end-of-life, misaligned with strategic direction, superseded by an Adopt-ring alternative, or carries unacceptable risk.	No new use cases. Existing deployments to be migrated per the Architecture Roadmap. Hold status is communicated to procurement to prevent new purchases.

Technology Radar Domains

A comprehensive enterprise Technology Radar is organized into domains. A typical domain structure for a large multinational:

Domain	Examples of Technologies Covered
Cloud Platforms & Infrastructure	AWS, Azure, GCP; containerization (Kubernetes, Docker); serverless platforms; cloud networking; CDN
Data Platforms & Analytics	Data warehouse platforms (Snowflake, BigQuery, Redshift); streaming (Kafka, Kinesis); BI tools (Power BI, Tableau); AI/ML platforms
Application Platforms & Frameworks	Programming languages; web frameworks; mobile development platforms; low-code/no-code platforms; CMS platforms
Integration & API	API gateway platforms; integration platforms (MuleSoft, Azure Integration Services); event streaming; ESB vs. event-driven patterns
Security & Identity	IAM platforms; SIEM tools; endpoint protection; secrets management; certificate management; zero-trust network access
DevOps & Operations	CI/CD platforms (GitHub Actions, Azure DevOps, Jenkins); infrastructure as code (Terraform, Pulumi); observability (Datadog, Dynatrace); incident management
Enterprise Applications	ERP platforms; CRM platforms; HCM platforms; finance platforms; procurement platforms
Collaboration & Productivity	Collaboration platforms (Microsoft 365, Google Workspace); video conferencing; document management; intranet platforms

Radar Governance Process

- Submission — any architect, engineer, or technology team can submit a technology for radar assessment using a structured submission template
- Assessment — the relevant domain architecture lead conducts a formal assessment covering: maturity, community support, vendor viability, security posture, total cost, skill availability, and strategic alignment
- Recommendation — the domain lead recommends a ring placement with rationale
- Review — the CEA and ARB review recommendations; significant placements (especially Adopt and Hold) require ARB endorsement
- Publication — the radar is published bi-annually as a visual artifact and as a searchable register in the Architecture Repository
- Communication — publication is accompanied by a change summary highlighting technologies that have moved between rings, with explanation and guidance

B.1.6 Architecture Roadmap

Purpose and Strategic Value

The Architecture Roadmap is the instrument through which the CEA translates architectural vision into investable, executable plans. It is the single most important artifact for connecting the EA practice to the organization's financial planning, program portfolio, and delivery machinery. A clear, credible roadmap is what separates an EA practice that influences organizational decisions from one that produces interesting documents.

The Architecture Roadmap serves as: the primary input to the annual technology investment planning process; the governing reference for program portfolio prioritization; the communication tool for demonstrating EA value to executives; and the baseline against which architecture progress is measured and reported.

CEA's Role

The Architecture Roadmap is one of the CEA's most personal deliverables. While the detailed content is assembled collaboratively across all architecture domains, the sequencing logic, the prioritization rationale, and the investment case are expressions of the CEA's architectural judgment. The CEA presents the roadmap to executive forums, defends its prioritization in investment committees, and is accountable for its alignment with business strategy.

Roadmap Architecture: Key Structural Elements

Work Packages

A work package is a discrete unit of architecture change — a coherent piece of work with a defined input state, output state, business benefit, and cost envelope. Work packages are the atomic building blocks of the roadmap.

Work Package Attribute	Description
Name	A clear, business-language name that communicates the change being made
Business Capability Enabled	Which BCM capability (or capabilities) this work package improves or enables
Description	2–3 sentences describing what will change as a result of this work package
Dependencies	Other work packages that must be completed before this one can start, or that cannot start until this one is complete

Work Package Attribute	Description
Business Benefit	The quantified or described business value: cost reduction, revenue enablement, risk reduction, compliance
Investment Estimate	Rough order of magnitude (ROM) capital and operating expenditure — not a precise project estimate
Risk	The primary architectural and delivery risks associated with this work package
Owner	The domain architect or domain lead responsible for this work package
Target Wave	Which roadmap wave this work package is assigned to

Transition Architectures

A transition architecture is a coherent, stable intermediate state of the enterprise architecture that the organization will operate for a defined period. Transition architectures are essential in multi-year transformations because they define 'rest stops' — points at which the organization can pause, consolidate, and extract value before the next wave of change.

- Each transition architecture must be a genuinely operable state — not a partially completed target that creates more problems than it solves
- Each transition architecture must deliver visible business value — not just 'plumbing' that enables future work
- Transition architectures should be described as a set of views across business, data, application, and technology domains, mirroring the full architecture structure

Roadmap Visualization

The Architecture Roadmap is most effective when presented as a visual timeline. Best-practice visualization includes:

- A horizontal timeline spanning 3–5 years, divided into waves or phases (e.g., Foundation, Core Modernization, Differentiation, Innovation)
- Work packages shown as horizontal bars on the timeline, color-coded by architecture domain (business, data, application, technology)
- Business capability outcomes shown as milestone markers — not technology deliveries
- A parallel track showing business events (regulatory deadlines, contract renewals, product launches) that anchor specific architecture deliveries

- An investment profile shown as a stacked bar or area chart below the timeline, showing capital and operating expenditure by year

Roadmap Governance

- Quarterly alignment — the roadmap is reviewed quarterly against actual delivery progress, business priority changes, and budget availability
- Annual refresh — a major roadmap refresh is conducted annually, aligned to the capital planning cycle, incorporating lessons from the past year and updated strategic priorities
- Change control — significant changes to roadmap sequencing or scope are reviewed by the ARB before implementation
- Dependency management — roadmap dependencies are actively tracked; slippage in one work package triggers an automatic review of downstream impacts

B.1.7 Architecture Decision Records (ADRs)

Purpose and Strategic Value

Architecture Decision Records are the institutional memory of the architecture. They capture not just what architectural decisions were made, but why — the context that made a decision necessary, the alternatives that were considered and rejected, and the consequences (positive and negative) that the decision entails. Without ADRs, the architecture knowledge is locked in the heads of individuals who may leave the organization — when they go, the rationale for every major technical decision goes with them.

ADRs also serve as a powerful governance tool. When a project team proposes a solution that contradicts a previously made architectural decision, the CEA can point to the ADR that documents the decision, its rationale, and its compliance implications — making the governance conversation factual rather than political.

CEA's Role

The CEA is the authority for enterprise-level ADRs — decisions that set constraints across the entire organization (e.g., 'We will use Azure as our primary public cloud provider'). The CEA reviews and endorses domain-level ADRs that have cross-domain implications. Solution-level ADRs are owned by solution architects but must be consistent with enterprise and domain ADRs — and must be registered in the Architecture Repository.

ADR Classification Levels

Level	Scope	Examples	Approval Authority
Enterprise ADR	Applies across the entire organization; sets constraints that all domains and projects must respect	Cloud provider strategy; enterprise integration pattern; master data management approach; identity platform selection	CEA
Domain ADR	Applies within a specific business or technology domain; consistent with Enterprise ADRs	CRM platform selection for Customer domain; data warehouse platform for Analytics domain; API gateway for Digital domain	Domain Architecture Lead + CEA review
Solution ADR	Applies to a specific project or solution; consistent with Enterprise and Domain ADRs	Message broker technology choice for Project X; caching strategy for Service Y; database engine for Application Z	Solution Architect; registered in Architecture Repository

Full ADR Structure

Field	Content Guidance	Common Mistakes
ID	Unique sequential identifier: ADR-ENT-001 (enterprise), ADR-CRM-003 (domain), ADR-PRJ-012 (solution)	Using non-unique or non-sequential IDs that make cross-referencing impossible
Title	Active-voice description of the decision made: 'Adopt Azure as the primary public cloud platform' not 'Cloud platform decision'	Titling it as a question ('Which cloud?') rather than a decision
Date	Date the ADR was formally accepted — not the date it was drafted	Recording draft date; ADRs sometimes take weeks to finalize
Status	Proposed → Accepted → Deprecated → Superseded (with link to superseding ADR)	Never updating status; leaving 'Proposed' ADRs in the register indefinitely
Deciders	Named individuals who made or endorsed the decision — CEA, ARB members, domain leads as appropriate	Leaving this blank, making it impossible to understand the authority behind the decision
Context	The situation that made a decision necessary: the business driver, the technical constraint, the competing options in play, and the criteria used to evaluate them. 2–5 paragraphs.	Writing only 1–2 sentences; context is the most valuable section for future readers
Decision	The choice that was made, stated unambiguously. 'We will adopt X' not 'We considered X'	Being vague: 'We will evaluate cloud options' is not a decision
Alternatives Considered	Each alternative that was seriously considered, with the reason it was rejected. No 'straw man' alternatives.	Listing alternatives without explaining why they were rejected
Consequences	Both positive consequences (what this decision enables) and negative consequences (what this decision forecloses, costs, or risks). Honest assessment.	Only listing positive consequences; omitting the costs and trade-offs
Compliance Notes	How projects will demonstrate compliance with this decision. What evidence is required in architecture compliance reviews?	Omitting this section, making governance of the decision impossible

Field	Content Guidance	Common Mistakes
Related ADRs	Links to ADRs this decision supersedes, depends on, or constrains	No cross-referencing; creating a disconnected set of isolated decisions

ADRs as Living Documents

ADRs are permanent records but not immutable ones. When circumstances change — a vendor is acquired, a technology reaches end-of-life, a better alternative emerges — the correct response is to create a new ADR that supersedes the old one, referencing it explicitly. Never delete or silently modify an accepted ADR. The history of decisions, including decisions that were later overturned, is part of the institutional knowledge the ADR system is designed to preserve.

Section B.2

Domain-Level Deliverables

The six artifact types produced by domain architects within specific business or technology domains

Domain-level deliverables bridge the enterprise architecture and the project level. They translate enterprise standards and direction into domain-specific architectural guidance — detailed enough to drive real design decisions within the domain, while remaining consistent with the principles, standards, and roadmap set by the central EA team.

The CEA's role with domain-level deliverables is one of governance and quality assurance rather than authorship. The CEA sets the metamodel (what each deliverable must contain and how it relates to other deliverables), reviews outputs for consistency with enterprise architecture, and ensures domain deliverables are registered in the Architecture Repository and actively used in domain investment and delivery decisions.

B.2.1 Domain Capability Maps

Purpose and Relationship to the BCM

Domain Capability Maps are the Level 3 and Level 4 elaborations of the Business Capability Model for a specific business domain. Where the enterprise BCM provides a view of capabilities at Level 1 and Level 2, Domain Capability Maps go deeper — providing the resolution needed to drive application portfolio decisions, process redesign, and technology investment within the domain.

A Domain Capability Map for the Customer domain, for example, might decompose 'Customer Acquisition' (a Level 2 BCM capability) into 8–12 Level 3 capabilities, and further decompose 'Campaign Management' into 4–6 Level 4 sub-capabilities — each mappable to specific applications, processes, and data entities.

CEA's Governance Role

- Mandate that every business domain has a Domain Capability Map maintained by the domain architect
- Ensure Domain Capability Maps are consistent with — and properly connected to — the enterprise BCM
- Review domain heat map assessments for consistency with enterprise investment priorities
- Use Domain Capability Maps as the primary input to domain portfolio and roadmap decisions

Structural Requirements

Element	Requirement
Scope	Clearly defined domain scope — which BCM Level 1 and Level 2 capabilities are elaborated in this map
Hierarchy	Minimum 3 levels of decomposition (Level 2 from BCM → Level 3 → Level 4 for complex capabilities); maximum 5 levels to maintain usability
Heat Map Assessment	Strategic importance and current performance assessed at Level 3 minimum; Level 4 where investment decisions require that resolution
Application Mapping	Each Level 3 capability mapped to the primary applications that support it; application count per capability shown
Data Entity Mapping	Key data entities consumed or produced by each Level 3 capability identified

Element	Requirement
Owner	Each capability has a named business owner (the person accountable for capability performance) and a named IT owner (the architect or system owner responsible for technology support)
Last Reviewed	Date of last formal review and next scheduled review; domain capability maps should be reviewed annually or on major strategy change

B.2.2 Domain Application Landscapes

Purpose

The Domain Application Landscape provides the detailed view of the application portfolio within a specific business or technology domain. Where the enterprise Application Portfolio provides a strategic, high-level view of all applications, the Domain Application Landscape provides the operational depth needed to plan domain-level transformation — showing not just what applications exist, but how they interact, where the integration complexity lies, and what the domain's specific rationalization opportunities are.

Key Contents

- Domain application inventory — all applications within scope, with full portfolio attributes (BCM mapping, TIME classification, TCO, health status)
- Application interaction diagram — visual representation of how applications within the domain exchange data, showing API dependencies, batch integrations, and shared data stores
- Integration complexity assessment — count and criticality of integrations per application; identification of high-integration-risk applications where replacement would be costly
- Gap analysis — capabilities in the Domain Capability Map that are under-served or unserved by current applications
- Domain rationalization plan — specific consolidation, retirement, and replacement recommendations with business cases
- Domain target application landscape — the future-state application map for the domain, consistent with enterprise platform decisions and the Architecture Roadmap

B.2.3 Domain Data Models

Purpose

Domain Data Models define the structure, relationships, and governance of the key data entities within a specific business domain. They bridge the enterprise data architecture (which defines data governance principles, master data management standards, and data platform strategy) and the solution level (where physical database schemas and data object models are designed).

Three Levels of Domain Data Modeling

Level	Name	Description	Primary Owner
Conceptual	Domain Conceptual Data Model	Identifies the major business data entities in the domain (Customer, Order, Product, Asset) and their key relationships. Technology-independent. Written in business language.	Business Domain Architect + Business SMEs
Logical	Domain Logical Data Model	Elaborates the conceptual model with attributes, data types, business rules, and cardinality. Still technology-independent. Defines master data entities and their authoritative sources.	Data Architecture Lead + Domain Architect
Physical	Physical Data Model	The technology-specific implementation of the logical model — database schemas, table structures, indexes. Produced at project level, not domain level.	Engineering Lead (not EA)

CEA Governance Requirements

- All enterprise master data entities (Customer, Product, Supplier, Asset, etc.) must be defined in a domain-level logical data model before any project designs a physical implementation
- Conflicts between domain data model definitions (e.g., two domains defining 'Customer' differently) are escalated to the CEA for resolution — this is a governance responsibility the CEA must personally engage with
- Domain data models are stored in the Architecture Repository and linked to the relevant Domain Capability Maps and Domain Application Landscapes

B.2.4 Domain Integration Architectures

Purpose

Domain Integration Architectures define the patterns, standards, and specific integration designs for how applications within and across domain boundaries exchange data and functionality. Integration is one of the most persistent sources of technical debt in large organizations — ungoverned, point-to-point integrations accumulate over years into a tightly coupled 'spaghetti' landscape that makes any change slow and risky.

By defining domain integration architectures, the CEA establishes the integration patterns that all projects within the domain must follow, preventing the accumulation of new integration debt even while the existing debt is being resolved.

Key Contents

- Integration principles for the domain — the specific integration standards applicable to the domain (e.g., 'All Customer domain integrations must use the enterprise API gateway; no direct database-to-database integrations permitted')
- Canonical data model — the agreed data formats for key entities exchanged across domain boundaries, ensuring that consuming systems do not need to understand the internal data structures of producing systems
- Integration pattern catalog — the approved integration patterns for the domain: synchronous REST APIs, asynchronous event streams, batch file transfers, and the criteria for choosing between them
- Current integration inventory — a complete map of existing integrations within and across the domain boundary, with health assessment and rationalization plan
- Target integration architecture — the future-state integration design for the domain, aligned to enterprise integration platform decisions
- API catalog — all domain APIs published or consumed, with versioning status, consumer count, and deprecation plan for legacy interfaces

B.2.5 Regional Architecture Variations

Purpose

In a large multinational, the enterprise architecture cannot be applied uniformly across all geographies. Legal, regulatory, cultural, language, infrastructure, and operational factors require regional adaptations. Regional Architecture Variations document the specific, formally approved deviations from enterprise standards that apply in a given geography, along with the rationale, governance approval, and remediation plan where relevant.

Common Drivers of Regional Variation

Driver	Architecture Impact	Examples
Data Sovereignty / Residency	Data for residents of certain jurisdictions may not leave the country or region; requires local data hosting	EU GDPR; China PIPL; Russia Federal Law 242-FZ; India PDPB
Regulatory Compliance	Sector-specific regulations may require specific technology controls, audit trails, or reporting architectures	Financial services regulations (MiFID II, Dodd-Frank); healthcare (HIPAA); critical infrastructure regulations
Local Market Requirements	Local payment methods, languages, tax regimes, or business processes require application-level adaptations	Local payment gateways (Alipay, PIX, UPI); multi-language support; local tax calculation engines
Infrastructure Limitations	Connectivity, power reliability, or cloud provider availability may differ significantly from enterprise standards	Limited cloud availability zones in certain markets; high-latency connectivity requiring local caching or edge processing
M&A Legacy	Recently acquired entities may have technology estates that cannot be immediately aligned to enterprise standards	Acquired company on a different ERP; different identity provider; different cloud platform

CEA Governance Requirements

- All Regional Architecture Variations must be formally approved by the CEA or the ARB — they are not informal exceptions
- Each variation must have a named owner (typically the regional architect), a documented rationale, and a remediation timeline where the variation is a temporary deviation rather than a permanent exception
- Regional variations are tracked in the Waiver Register and reviewed annually to assess whether the conditions that justified the variation still apply

B.2.6 Domain Roadmaps

Purpose

Domain Roadmaps provide the domain-level elaboration of the enterprise Architecture Roadmap. Where the enterprise roadmap shows major waves of transformation at a strategic level, domain roadmaps show the specific sequence of application changes, capability improvements, and integration rationalization within a domain that are needed to deliver the enterprise waves.

Relationship to the Enterprise Architecture Roadmap

Domain Roadmaps are nested within the enterprise Architecture Roadmap. The enterprise roadmap defines the waves and work packages at an enterprise level; Domain Roadmaps decompose each enterprise work package into the specific domain-level initiatives needed to deliver it. A change to the enterprise roadmap (a wave brought forward, a work package descoped) requires a corresponding update to the relevant Domain Roadmaps.

Key Contents

- Domain transformation waves — the sequence of major changes within the domain, aligned to enterprise roadmap waves
- Application disposition timeline — when each 'Migrate' or 'Eliminate' application in the domain portfolio will be addressed, in what sequence
- Capability improvement trajectory — for each underperforming capability, the planned improvement initiatives and their expected impact on the heat map assessment
- Integration rationalization plan — the sequence in which legacy integrations will be replaced with enterprise standard patterns
- Domain investment profile — capital and operating expenditure requirements by year for the domain transformation
- Domain dependencies — explicit dependencies on enterprise programs, other domain roadmaps, or external factors (vendor releases, regulatory deadlines)

Section B.3

Project-Level Deliverables

The six artifact types produced by solution architects on specific delivery programmes

Project-level deliverables are produced by solution architects embedded in or assigned to specific delivery programs. They translate enterprise and domain architectural standards into the concrete designs that engineering teams build from. The CEA's role at the project level is principally one of governance — ensuring that project-level deliverables are produced, that they conform to enterprise and domain standards, and that significant deviations are identified and managed through the formal waiver process.

The quality of project-level deliverables is the ultimate test of the EA practice's effectiveness. Perfectly crafted enterprise architecture that is ignored at project level has zero organizational value. The CEA must design the governance process — Architecture Contracts, compliance reviews, and the ARB — to ensure that project-level architecture is genuinely connected to enterprise direction.

B.3.1 Solution Architecture Document (SAD)

Purpose and Strategic Value

The Solution Architecture Document is the primary architecture deliverable for a delivery project. It translates the architectural requirements in the Architecture Contract into a specific, implementable design that satisfies the project's functional and non-functional requirements while conforming to enterprise and domain architecture standards.

The SAD serves three audiences simultaneously: the delivery team, who builds from it; the EA function, who reviews it for compliance; and future architects, who need to understand what was built and why.

CEA's Governance Role

The CEA does not author SADs — that is the solution architect's responsibility. The CEA's governance role is to:

- Ensure every significant project has an Architecture Contract that specifies SAD requirements
- Ensure a domain architect reviews every SAD for compliance with enterprise and domain standards before build begins
- Escalate significant non-conformances to the ARB for resolution
- Use patterns of SAD non-conformance as signals for where enterprise standards may need to be updated or better communicated

Detailed SAD Structure

Section	Content	Quality Standard
1. Executive Summary	What the solution does; what problem it solves; what the key architectural decisions are; what the primary risks are. Maximum 2 pages. Written for a non-technical senior stakeholder.	A project sponsor with no technical background can read this section and understand what is being built and why.
2. Business Context	The business capability(ies) this solution supports (referenced to the BCM); the business requirements driving the solution; the strategic alignment to the Architecture Roadmap. References the Architecture Contract.	Every business requirement is traceable to a named business capability and a named business stakeholder.

Section	Content	Quality Standard
3. Architecture Constraints and Standards	The enterprise and domain standards, principles, and ADRs that this solution must comply with. The Architecture Contract reference. Any approved waivers.	Complete reference to all applicable standards; no standards omitted or unknown.
4. Solution Overview	A high-level description of the solution and its major components. A context diagram showing the solution in its environment — what systems, users, and external parties it interacts with.	A C4 Context Diagram or equivalent. Non-technical stakeholders can understand the solution boundary from this section.
5. Functional Architecture	The decomposition of the solution into functional components; what each component does; how components interact. A component diagram. Application of domain integration patterns.	C4 Container and Component diagrams or equivalent. All integration patterns reference enterprise standards.
6. Data Architecture	What data the solution creates, reads, updates, and deletes; data model (logical); master data entities and their authoritative source; data residency and sovereignty compliance.	Data entities mapped to the domain data model. Master data sourced from authoritative systems, not replicated without governance approval.
7. Infrastructure and Deployment Architecture	Where the solution runs (cloud region, platform, service tier); how it is deployed (containerized, serverless, IaaS); networking and security architecture; high-availability and disaster recovery design.	Deployment architecture consistent with enterprise cloud standards and Technology Radar Adopt-ring technologies.
8. Security Architecture	Authentication and authorization design; data protection controls; API security; secrets management; security testing approach. Aligned to enterprise security architecture standards.	Security architecture reviewed by the Security Architecture Lead or CISO function before build begins.
9. Non-Functional Requirements	Performance, availability, scalability, recoverability, and compliance requirements — as specific, measurable statements. See NFR Specification below.	Every NFR has a measurable target and a testing approach. 'Highly available' is not a valid NFR statement.
10. Architecture Decision Records	All significant architectural decisions made during solution design, in ADR	Every significant design choice — technology selection, integration pattern, data storage

Section	Content	Quality Standard
	format. Cross-referenced to enterprise and domain ADRs where relevant.	approach — documented as an ADR.
11. Risks and Mitigations	The architectural risks associated with this solution — technical complexity, integration risk, vendor dependency, scalability uncertainty — each with a mitigation approach and residual risk assessment.	No risk rated 'High' without an active mitigation in place or an explicit acceptance decision by the appropriate authority.
12. Architecture Compliance Statement	A completed compliance checklist demonstrating adherence to all requirements in the Architecture Contract. Any non-conformances identified, with waiver status.	Completed by the solution architect and reviewed by the domain architect. Signed off before build begins.

B.3.2 Architecture Compliance Assessment

Purpose

The Architecture Compliance Assessment is the formal record of a review conducted by the EA function to assess whether a solution's architecture conforms to the standards, principles, and requirements defined in the Architecture Contract. It is the primary mechanism through which the CEA ensures that what gets built matches what was designed and approved.

Assessment Process

Stage	Activity	Owner
Submission	The delivery team submits the SAD, architecture diagrams, and a completed compliance self-assessment checklist to the assigned domain architect	Solution Architect
Desk Review	The domain architect reviews the submission against the Architecture Contract and enterprise standards prior to the review meeting	Domain Architect
Review Meeting	A structured meeting (typically 1–2 hours) where the solution architect presents the architecture and the domain architect asks clarifying questions and raises concerns	Domain Architect + Solution Architect
Finding Classification	Each finding is classified: Compliant, Non-Conformant (Minor), Non-Conformant (Major), or Waiver Required	Domain Architect
Report Production	The domain architect produces the formal Compliance Assessment Report within 5 business days of the review meeting	Domain Architect
Resolution Tracking	Non-conformances are entered into the remediation tracker; waivers are submitted to the ARB; the domain architect tracks resolution to closure	Domain Architect + EA Practice Manager

Compliance Assessment Report Structure

- Assessment summary — overall compliance status (Compliant, Conditionally Compliant, Non-Conformant), date, project, solution architect, reviewing architect

- Scope of assessment — which standards, principles, and ADRs were assessed; which were deemed not applicable and why
- Findings register — each finding with: standard referenced, conformance status, evidence reviewed, description of non-conformance (if applicable), recommended resolution
- Waiver requests — any non-conformances for which a waiver is being requested, with business justification
- Remediation plan — for each non-conformance, the planned resolution, responsible party, and target date
- Sign-off — domain architect signature; CEA or ARB sign-off for significant non-conformances or waivers

B.3.3 High-Level Design (HLD)

Purpose and Distinction from the SAD

The High-Level Design elaborates the Solution Architecture Document into a design that is sufficiently detailed for engineering teams to begin detailed technical design and build. Where the SAD defines what the solution will do and what architectural patterns it will follow, the HLD defines how it will be built at a system-component level.

The HLD is the primary handoff document from solution architecture to engineering. It is owned by the solution architect but must be sufficiently detailed that an engineering lead who was not involved in the architecture phase can pick it up and build from it.

Key Content Requirements

Section	Content
Component Architecture	Detailed component diagram showing all application components, their responsibilities, and their interactions. Technology decisions at component level (programming language, framework, runtime).
Data Design	Logical data model for solution-specific data; integration with master data; data migration approach (if replacing existing system); data archiving and retention approach.
Integration Design	For each integration point: the integration pattern (synchronous API, async event, batch), the data exchanged (with canonical data model reference), the protocol and format, the error handling approach, and the SLA.

Section	Content
Infrastructure Design	Platform services used (cloud services, container platform, database service); infrastructure sizing (initial capacity and scaling model); environment strategy (development, test, staging, production).
Security Design	Detailed authentication and authorization design (identity provider integration, role model, permission matrix); data encryption at rest and in transit; secrets management implementation; API security controls.
Operational Design	Monitoring and alerting design (what metrics, what thresholds, what alerting channels); logging design (what events, retention, access); backup and recovery design; incident escalation path.
Deployment Design	CI/CD pipeline design; deployment strategy (blue-green, canary, rolling); environment promotion process; release management approach.

B.3.4 Low-Level Design (LLD)

Purpose

The Low-Level Design translates the HLD into the detailed technical specifications used by individual engineers to implement specific components. LLDs are component-scoped — each significant component or service has its own LLD. They are produced by engineering leads rather than solution architects, though the solution architect reviews them for architectural consistency.

CEA's Role

The CEA has minimal direct involvement with LLDs. The CEA's governance interest is confined to ensuring that: the LLD production process exists and is followed; LLDs are reviewed by the solution architect before build begins; and significant deviations from the HLD surfaced during LLD development are escalated to the domain architect. LLDs are technical implementation documents — they should not require CEA review unless they reveal a fundamental architectural problem.

Key Content Requirements

- Detailed class/module/service design — internal structure, interfaces, dependencies, and data flows within the component
- Physical data model — database schema, table definitions, indexes, and constraints for data stores owned by this component

- API specification — full OpenAPI or AsyncAPI specification for all interfaces exposed or consumed by the component
- Error handling and resilience patterns — timeout values, retry logic, circuit breaker configuration, dead letter queue design
- Performance design — caching strategy, connection pooling, query optimization approach, and expected performance under load
- Unit and integration test approach — what will be tested, how, and what the coverage targets are

B.3.5 Non-Functional Requirements (NFR) Specification

Purpose

The NFR Specification defines the quality attributes that the solution must achieve — the 'ilities' that determine whether the solution is fit for purpose in production. NFRs are systematically under-specified in most delivery projects, leading to solutions that work in testing but fail in production at scale or under adversarial conditions.

The CEA establishes enterprise NFR standards — baseline quality attributes that all solutions must meet — and the NFR Specification applies and extends these baselines for the specific solution context.

NFR Categories and Specification Requirements

NFR Category	Key Attributes	Specification Guidance
Performance	Response time (average and 95th/99th percentile); throughput (transactions per second); latency under peak load	Specify for each key user journey and API endpoint separately. Peak load = 3× average load unless evidence suggests otherwise. Target must be measurable in a load test.
Availability	Uptime SLA (e.g., 99.9% = 8.7 hours downtime per year); planned maintenance window; RTO (Recovery Time Objective); RPO (Recovery Point Objective)	SLA must be tiered by criticality (Tier 1: 99.95%; Tier 2: 99.9%; Tier 3: 99.5%). RTO and RPO must be stated in hours, not generalities.
Scalability	Maximum concurrent users; peak transaction volume; data volume	Specify both horizontal and vertical scaling approach. Define the load level at which

NFR Category	Key Attributes	Specification Guidance
	growth rate; auto-scaling triggers and limits	architecture review is required before further scaling.
Security	Authentication standards; authorization model; data classification and protection requirements; penetration testing requirements; vulnerability scanning frequency	All solutions processing Confidential or Restricted data must undergo penetration testing before go-live. OWASP Top 10 compliance is mandatory for all web-facing components.
Compliance	Regulatory frameworks applicable to this solution (GDPR, PCI-DSS, HIPAA, SOX, etc.); audit logging requirements; data retention and deletion obligations	Specify the regulatory control that applies and the architectural evidence required to demonstrate compliance.
Operability	Monitoring coverage (% of components monitored); alerting thresholds; log retention period; support hours; incident response SLA	All production components must emit standard observability data (metrics, traces, logs) to the enterprise observability platform.
Recoverability	Disaster recovery strategy (active-active, active-passive, backup-restore); failover time; data backup frequency and retention	Disaster recovery approach must be consistent with the solution's Tier classification. DR tests must be conducted annually for Tier 1 and Tier 2 solutions.
Maintainability	Code coverage targets; technical debt tolerance; documentation requirements; dependency update cadence	No new dependency with a known critical CVE may be introduced. DORA metrics targets (deployment frequency, lead time, MTTR) specified for each solution.

B.3.6 Integration Design Document

Purpose

The Integration Design Document provides the detailed specification of each integration between the solution and other systems. It is the binding contract between the solution

team and the teams responsible for the systems it integrates with — defining precisely what data is exchanged, in what format, at what frequency, with what error handling, and with what SLA.

Integration is the most common source of project delay and post-deployment defects in large-scale enterprise programs. Rigorous Integration Design Documents — reviewed by the domain architect before build begins — are one of the highest-value governance interventions the EA function can make.

Integration Design Document: Per-Integration Structure

Section	Content
Integration ID and Name	Unique identifier and descriptive name: INT-CRM-001: Customer Master to Billing System
Integration Pattern	The approved pattern: synchronous REST API, asynchronous event (specify broker and topic), batch file transfer (specify format and schedule), or database-to-database (must have waiver if not enterprise standard)
Producer and Consumer	The system producing the data/event and the system consuming it; named integration owners for each system
Data Payload	Complete specification of the data exchanged: field names, data types, mandatory/optional status, business rules, and reference to canonical data model where applicable
Trigger and Frequency	What triggers the integration: user action, scheduled batch, event, or threshold; frequency (real-time, near-real-time, hourly, daily)
Volume and Performance	Expected message volume at average and peak load; maximum acceptable latency for synchronous integrations; maximum acceptable lag for asynchronous integrations
Error Handling	What happens when the integration fails: retry logic (count, backoff strategy), dead letter queue design, alerting and notification, manual intervention process
Security	Authentication between systems (mTLS, OAuth 2.0, API key); data encryption in transit; access control — who or what can call this interface
Versioning and Change Control	API versioning strategy; how consumers will be notified of changes; backward compatibility commitments; deprecation timeline
Testing Approach	How the integration will be tested: consumer-driven contract testing, integration test environment, test data strategy

Section	Content
Monitoring and SLA	What metrics will be monitored (latency, error rate, volume); what thresholds trigger alerts; what the SLA is; who is notified on breach

Addendum B Summary: Deliverable Quality Framework

The following framework provides the CEA with a structured way to assess and improve the quality of the architecture deliverables across all three tiers.

Deliverable	Signs of High Quality	Signs of Poor Quality	CEA Action
Architecture Vision	Business leaders quote it in strategy discussions; guides investment decisions; updated within 90 days of strategy changes	Sits on SharePoint unread; full of technology diagrams; last updated 18 months ago	Rewrite with business language; co-author with a business executive; present quarterly
Architecture Principles	Referenced in ARB decisions; project teams cite them; violations are flagged immediately	Nobody can name them; they are never violated because nobody follows them	Reduce to 10 strongest principles; workshop with senior leaders; build into ARB template
Business Capability Model	Used in investment committee; drives application portfolio decisions; owned by business leaders	Only architects have seen it; organizational units used instead of capabilities; never updated	Run BCM workshop with COO and business unit heads; make it a living portfolio tool
Application Portfolio	Portfolio health metrics reported quarterly; TIME classification drives budget decisions; count is declining	Count is unknown or growing; no TIME classification; TCO data missing	Commission application discovery; build TCO model; present rationalization business case to CFO
Technology Radar	Teams consult it before technology selections; Hold-ring technologies are being retired; updated bi-annually	Nobody knows it exists; projects use whatever they like; last update was 3 years ago	Relaunch with engineering community involvement; integrate with procurement approval process
Architecture Roadmap	Shapes the annual investment plan; business leaders can describe the	Exists in a slide deck; no dependency tracking; not aligned to budget cycle	Rebuild with work package structure; align to capital

Deliverable	Signs of High Quality	Signs of Poor Quality	CEA Action
	waves; progress tracked quarterly		planning; publish progress dashboard
ADRs	Searchable repository; project teams reference enterprise ADRs; history of decisions preserved	Decisions made verbally in meetings and forgotten; same question asked repeatedly	Mandate ADR production for all significant decisions; build into Architecture Contract template
Domain Capability Maps	Domain architects maintain them actively; used in domain investment planning	Exist as one-time artifacts from an old project; never updated	Assign ownership to domain architects; include in EA governance KPIs
Solution Architecture Documents	Written before build starts; compliance assessment completed; engineering teams build from them	Written retrospectively to satisfy governance; too vague to guide build; ignored after approval	Enforce Architecture Contract gate before build approval; provide SAD template and training
NFR Specifications	Measurable targets for all quality attributes; tested in load and security tests before go-live	Vague statements like 'highly available' and 'good performance'; never tested	Publish enterprise NFR baseline standards; include NFR review in compliance assessment

References for This Addendum

[1] The Open Group. TOGAF Standard, Version 9.2. The Open Group, 2018. opengroup.org/togaf — Primary reference for architecture deliverable definitions, the Architecture Content Framework, and the Architecture Repository.

[2] The Open Group. TOGAF Series Guide: Business Capabilities. The Open Group, 2020 — Detailed guidance on Business Capability Model construction, assessment, and governance.

[3] Ross, J.W., Weill, P. & Robertson, D.C. Enterprise Architecture as Strategy. Harvard Business School Press, 2006 — Foundation for Business Capability Model approach and operating model design.

[4] Gartner. TIME Application Portfolio Analysis Model. Gartner Research, 2014. gartner.com — Defines the Tolerate, Invest, Migrate, Eliminate framework for application portfolio management.

[5] Gartner. PACE Layered Application Strategy. Gartner Research, 2012. gartner.com — Systems of Record, Differentiation, and Innovation model for application portfolio segmentation.

[6] ThoughtWorks. Technology Radar (bi-annual publication). thoughtworks.com/radar — The original and most widely referenced technology radar; the primary model for enterprise technology standards governance.

[7] Nygard, M.T. Release It! Design and Deploy Production-Ready Software, 2nd ed. Pragmatic Programmers, 2018 — Essential reference for Non-Functional Requirements specification and resilience patterns.

[8] Richards, M. & Ford, N. Fundamentals of Software Architecture. O'Reilly, 2020 — Comprehensive reference for architecture decision-making, ADR practice, and quality attribute (NFR) design.

[9] Hohpe, G. & Woolf, B. Enterprise Integration Patterns. Addison-Wesley, 2003. enterpriseintegrationpatterns.com — The canonical reference for integration design patterns referenced in Integration Design Documents.

[10] C4 Model. Simon Brown. c4model.com — The C4 (Context, Container, Component, Code) model used for Solution Architecture Document diagrams and HLD component diagrams.

[11] Brown, S. Software Architecture for Developers. Leanpub, 2019 — Practical guidance for producing Solution Architecture Documents, including the C4 model in detail.

[12] NYGARD, M. 'Architecture Decision Records.' Cognitect Blog, 2011 — The original post defining the ADR format, widely adopted across the industry.

[13] Fowler, M. 'Sacrificial Architecture.' martinfowler.com, 2014 — Important context for Architecture Roadmap design and the value of transition architectures.

[14] OWASP Foundation. OWASP Top 10. owasp.org — The mandatory security reference for Solution Architecture Document security sections and NFR security specifications.

[15] DORA Research Program. Accelerate: State of DevOps Report. dora.dev — The primary reference for maintainability NFR metrics (deployment frequency, lead time for changes, MTTR, change failure rate).

End of Addendum B — CEA Deliverables: A Complete Practitioner's Reference

Append to: Chief Enterprise Architect Practitioner's Guide, Version 1.0 | Following Addendum A