



Grounded Architecture

Redefining IT Architecture Practice in the Digital Enterprise

Željko Obrenović

Grounded Architecture

Redefining IT Architecture Practice in the
Digital Enterprise

Željko Obrenović

This book is available at <https://leanpub.com/groundedarchitecture>

This version was published on 2025-05-26



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2025 Željko Obrenović

Contents

1: Introduction	1
1.1: Grounded Architecture Overview	4
1.2: What Will You Learn?	9
1.3: Beyond Rigid Frameworks: IT Architecture as a Collaborative, Adaptable, Outcome-Oriented Practice	12
1.4: Key Influences	15
1.5: Why This Book?	17
1.6: A Part of the Bigger Picture: A Trilogy in Four Parts	19
1.7: A Bit of Personal History	21
1.8: Stay Connected	23
1.9: Acknowledgments	24
2: Context: Fast-Moving Global Organizations	25
2.1: Global Scale	27
2.2: Multi-Dimensional Diversity	29
2.3: Nonlinear Growth Dynamics	32
2.4: Synergy and Transformation Pressures	34
2.5: Decentralization and Loose Coupling	36
2.6: Questions to Consider	38
3: Goals: Adapting, Growing, and Using Data	39
3.1: Goal 1: Executing At Scale	41
3.2: Goal 2: Adaptivity	42
3.3: Goal 3: Enhancing Decision-Making Quality with Data	43
3.4: Goal 4: Maximizing Organizational Alignment	44
3.5: Goal 5: Maximizing Organizational Learning	45
3.6: Questions to Consider	46

CONTENTS

Part I: Grounded Architecture Framework	47
4: Grounded Architecture Framework: Foundations	48
5: Lightweight Architectural Analytics	51
5.1: Why I Start With Data:	54
5.2: Examples of Lightweight Architectural Analytics Tools	56
5.3: Requirements for Lightweight Architectural Analytics	68
5.4: Tips for Building Lightweight Architectural Analytics	71
5.5: Tips for Using Lightweight Architectural Analytics	74
5.6: To Probe Further	77
5.7: Questions to Consider	78
6: Collaborative Networks	79
6.1: Background: Centralized vs. Federated Architecture Practice	83
6.2: The Hybrid Model	85
6.3: Tips for Building Collaborative Networks	89
6.4: To Probe Further	92
6.5: Questions to Consider	93
7: Operating Model: General Principles	94
7.1: Examples of Architecture Activities	97
7.2: Guiding Principles for Architectural Excellence: Policies, Autonomy, and Engagement	100
7.3: Embracing Diversity	107
7.4: Setting Boundaries	109
7.5: To Probe Further	112
7.6: Questions to Consider	113
8: Cooperation-Based Operating Model: Six Simple Rules	114
8.1: Background: Hard and Soft Management	116
8.2: Collaboration Approach	120
8.3: Rule 1: Understand What Your People Do	122
8.4: Rule 2: Reinforce Integrators	124
8.5: Rule 3: Increase the Total Quantity of Power	126
8.6: Rule 4: Increase Reciprocity	128
8.7: Rule 5: Extend the Shadow of the Future	130
8.8: Rule 6: Reward Those Who Cooperate	132
8.9: To Probe Further	134

8.10: Questions to Consider	135
9: Governance: Nudge, Taxation, Mandates	136
9.1: Nudging as a Governance Tool	140
9.2: Taxation (Economic Incentives) as a Governance Mechanism	143
9.3: Mandates and Bans as Governance Tools	146
9.4: Questions to Consider	149
10: Transforming Organizations with Grounded Architecture	150
10.1: Executing at Scale	152
10.2: Adaptivity	155
10.3: Improving the Quality of Decision-Making with Data	158
10.4: Maximizing Organizational Alignment	161
10.5: Maximizing Organizational Learning	163
10.6: Questions to Consider	166
11: Next Steps: Leveraging The Potential of Generative AI	167
11.1: What Generative AI Can Do for Today's Architect	170
11.2: Integrating GenAI with Grounded Architecture: Connecting Capabilities to the Framework	172
11.3: Practical Examples: GenAI in Action	175
11.4: Balancing Benefits, Risks, and Challenges	179
11.5: Best Practices for Responsibly Adopting GenAI	183
11.6: The Evolving Landscape	186
11.7: Questions to Consider	189
11.8: Tables	190
11.9: To Probe Further: References	195

1: Introduction



image by rasica from istock

IN THIS SECTION, YOU WILL: Understand what this book is about and how to use it.

KEY POINTS:

- This book will share my approach to running an IT architecture practice in larger organizations based on my experience at AVIV Group, eBay Classifieds, and Adevinta. I call this approach “Grounded Architecture”—architecture with strong foundations and deep roots.
- Prioritizing people interactions and data over processes and tools, Grounded Architecture aims to connect an architecture practice to all organizational levels as an antidote to the “ivory tower” architecture.
- I also explain my motivation to write this book.

Have you ever wondered how to run an IT architecture practice without **feeling isolated in an ivory tower**, detached from your organization’s people, problems, and realities?

If so, this book is for you.

Based on my experiences at AVIV Group, eBay Classifieds, and Adevinta, I introduce an approach I call **Grounded Architecture**. This **practical, human-centered method** emphasizes connecting to real-world challenges, real-time data, and real people.

Grounded Architecture is about **keeping your feet on the ground** rather than floating above the fray. It focuses on **making architecture relevant, collaborative, and responsive** to our fast-paced environments. It redefines architecture not as a set of abstract frameworks but as a living, evolving discipline rooted in relationships, results, and continuous learning.

In too many organizations, architects are viewed as distant wizards casting spells from a tower—producing documents no one reads and models no one uses. Grounded Architecture challenges that perception by **promoting an architecture practice embedded in the organization’s fabric**, which is connected to strategy, delivery, operations, and, most importantly, people.

This approach places data and collaboration at the center rather than relying on heavy-handed processes or over-engineered tools. It **draws**

inspiration from Grounded Theory, a research methodology that develops theories from real-world data rather than abstract ideals. In the same spirit, Grounded Architecture evolves through continuous feedback, real usage, and tangible impact—not just best practices from outdated textbooks.

Throughout this book, I will share insights, tools, and stories to help you build an architecture practice that is:

- **Collaborative**, not isolated
- **Adaptable**, not rigid
- **Outcome-oriented**, not obsessed with processes

If you’re an architect, engineering leader, or technologist looking to make architecture more effective and human, this book offers a grounded path forward.

1.1: Grounded Architecture Overview

This book introduces **Grounded Architecture**, a practical, people-centered approach to managing an IT architecture practice—especially in complex, fast-paced organizations. It is based on my experiences at **AVIV Group**, **eBay Classifieds**, and **Adevinta**, and is designed to help you overcome the stereotype of the “**architect in the ivory tower**.”

Instead, this approach reimagines architecture as **deeply embedded**, **collaborative**, and **realistic**—firmly rooted in the needs of teams, data, and outcomes.

1.1.1: Two Main Parts

The book is divided into two complementary sections (see *Figure 1*):

1. **Framework** – The essential components that will enable you to build your own Grounded Architecture practice.
2. **Guiding Principles** – Practical insights, tips, and inspiration to help you apply the framework effectively and sustainably.

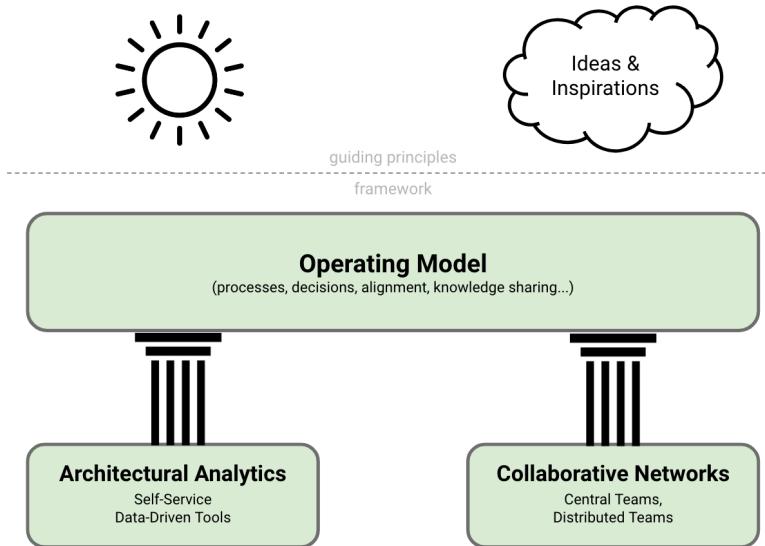


Figure 1: Grounded Architecture Overview

1.1.2: The Grounded Architecture Framework

The framework consists of **three core elements**:

- **Lightweight Architectural Analytics (Data):** This element enables data-informed decisions by providing accurate, up-to-date insights into your organization's technology landscape.
- **Collaborative Networks (People):** This emphasizes the power of relationships and peer networks to scale architectural influence and ensure shared ownership of key decisions.
- **Operating Model:** This offers the governance structures, principles, and practices that shape how architecture functions across the organization. This includes:
 - **General Principles:** Aligning architecture with your organization's reality by focusing on collaboration, autonomy, and strategic alignment rather than top-down control.

- **IT Governance: Nudge, Taxation, Mandates:** Governance should be a balanced mix of soft influence (nudging), economic signals (taxes), and clear boundaries (mandates/bans).
- **Leveraging Generative AI within Grounded Architecture Framework¹:** Based on my early explorations, this section shares practical ways to integrate GenAI tools into architectural workflows while maintaining integrity and oversight.

These components together form a pragmatic, adaptive foundation for modern architectural practice—grounded in data, people, and results.

1.1.3: Guiding Principles and Inspiration (Books 2 and 3)

Beyond the framework, this book offers a set of practical tools and reflections that explore the human, strategic, and behavioral dimensions of architecture. These principles are grouped into thematic areas:

1.1.3.1: On Being an Architect

Explore the mindset and meta-skills of effective architects:

- Building Skills, Making Impact, Leadership
- Thinking Like an Architect: Superglue
- Balancing Curiosity and Skepticism
- Navigating Career Paths

1.1.3.2: On Human Complexity

Architecture is not just technical—it is deeply social:

- The Culture Map
- The Human Side of Decision-Making
- Cooperation-Based Organizations
- Effortless Architecture

¹gen-ai

1.1.3.3: On Strategy²

Position architecture as a strategic enabler:

- Enterprise Architecture as Strategy³
- Outsourcing Strategies⁴

1.1.3.4: Expanding the Architect's Toolkit

Learn from adjacent fields to elevate your practice:

- Economics and ROI Modeling
- Customer-Centric Thinking
- Decision Intelligence and Systems Thinking
- Lessons from Mega-Projects

1.1.4: Online Appendix: My Architecture Backpack

Throughout my career, I have relied on a curated set of resources—books, tools, and articles—that continue to shape my thinking. I've compiled these for you in the [appendix](#), which includes:

- Favorite Quotes⁵
- Bookshelf⁶
- Growing & Hiring Architects⁷
- Effective Communication⁸
- Dealing with Toxic Colleagues⁹
- Scapegoating at Work¹⁰

²[strategy](#)

³[ea-as-strategy](#)

⁴[outsourcing](#)

⁵<https://grounded-architecture.io/quotes>

⁶<https://grounded-architecture.io/bookshelf>

⁷<https://grounded-architecture.io/growing>

⁸<https://grounded-architecture.io/communication>

⁹<https://grounded-architecture.io/toxic-colleagues>

¹⁰<https://grounded-architecture.io/scapegoating>

- ISO 25010 Quality Model¹¹
- Cloud Design Patterns¹²
- Business Architecture Insights¹³
- Value-Based Strategy¹⁴
- Digital Marketplace Thinking¹⁵
- Connecting Marketing and Architecture¹⁶
- Culture as a Strategy¹⁷

Before diving into the full framework, I would like to share some thoughts on **why** I wrote this book.

¹¹<https://grounded-architecture.io/iso25010>

¹²<https://grounded-architecture.io/cloud-design-patterns>

¹³[business-architecture](#)

¹⁴[value-based-strategy](#)

¹⁵[marketplaces](#)

¹⁶[marketing-sales-strategy](#)

¹⁷[culture-strategy](#)

1.2: What Will You Learn?

This book is a **practical guide** to building and managing a **robust, modern IT architecture practice**. It aims to help you align with organizational goals, navigate complexity, and thrive in today's rapidly evolving IT environments.

Whether you're a practicing architect, a leader of architects, or someone working closely with them, this book will provide you with **insight, structure, and inspiration**.

1.2.1: Key Topics

You'll discover useful perspectives and actionable advice on how to:

- Create **flexible organizational and technical structures** that support and scale IT architecture work.
- Define **architecture roles, skills, and career paths** to help your architects grow and succeed.
- Operate an architecture practice effectively in diverse, complex, and multicultural environments.
- Learn from **real-world lessons and insights** based on my experiences at AVIV Group, eBay Classifieds, and Adevinta.

1.2.2: Format & Structure

I've organized this book as a **high-level “playbook”** for running an architecture practice or working as an architect. Each section introduces a topic, offers **concrete tips**, and concludes with **reflective questions** to help you apply the ideas in your context.

You can read this book **cover to cover** or dip into it based on your current needs and interests. With rich **illustrations** and conceptual visuals, it also functions as a **“coffee table book”**—perfect for team discussions, sparking conversations, or flipping through for inspiration.

1.2.3: What This Book Is Not

This book is **not a technical manual**. It does not discuss cloud infrastructure, cybersecurity, database sharding, or performance optimization in depth. While these are essential skills, they are well-covered elsewhere.

Instead, this book broadens your **architectural perspective**, helping you apply your technical knowledge to **complex organizational ecosystems**. It's written for both **hands-on architects** and those managing or supporting them.

1.2.4: Is This a Proven Method?

This book isn't a prescriptive methodology, nor is it scientifically validated. It is a **personal and opinionated guide**, grounded in day-to-day experience rather than academic theory.

What I present here has been tested across **three major companies**, and while every organization is different, I believe the principles are **general enough to be reusable and practical enough to be valuable**.

In a field filled with frameworks and formal models, **real stories and tested approaches** often provide the most valuable guidance. I also encourage others to share their experiences because our discipline becomes stronger through **practical reflection and honest exchange**, rather than abstract theory.

1.2.5: Who Should Read This Book?

This book is intended for:

- **IT architects** who want to enhance their impact, influence, and clarity.
- **Engineering leaders** who manage architects and wish to organize their work more effectively.
- **Business and product leaders** looking to understand the value of architecture and how to collaborate better with technical teams.

I designed it to be **accessible to both technical and non-technical readers**, and useful whether you're building a practice from scratch or refining an existing one.

1.2.6: Applying Ideas in Practice: Architecture as Cooking

I like to think of starting or improving an architecture practice as a form of **culinary art**.

Imagine arriving at a new company like a chef stepping into an unfamiliar kitchen, carrying a suitcase filled with your favorite tools and spices. Your most important work comes next: discovering the **local ingredients**.



image by hispanolistic from istock

Great cooking depends on what's fresh and available, but great architecture relies on **your organization's people, culture, and context**. Your frameworks and experience provide structure, but the authentic flavor comes from your team: their skills, insights, and unique organizational knowledge.

While I'll share recipes, methods, and even a few secret sauces, remember: **you'll always need to adapt to your local ingredients**.

1.3: Beyond Rigid Frameworks: IT Architecture as a Collaborative, Adaptable, Outcome-Oriented Practice

In this book, I use the terms **IT architecture** and **IT architects** broadly and inclusively. I do not draw strict lines between enterprise, solution, technical, or software architecture, nor do I narrowly define titles such as enterprise architect, solution architect, or staff-level technical leaders like principal engineers.

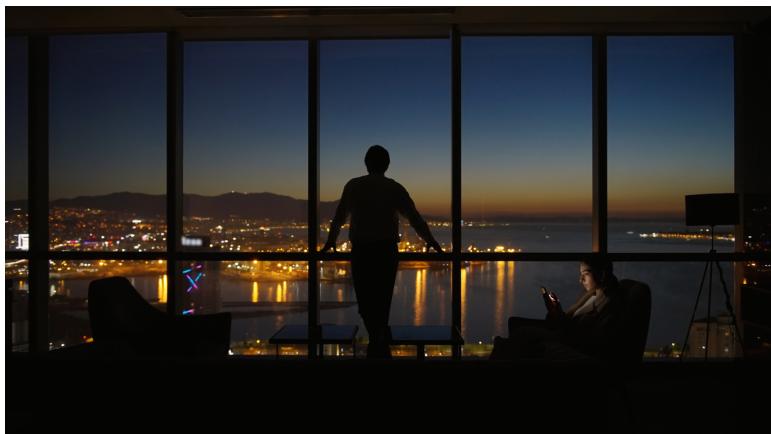


image by brightstars from istock

Instead, I view **IT architecture** as a **cohesive function**—a bridge that connects technology, business, product, and organizational strategy. Architecture encompasses more than a set of tasks or responsibilities; it serves as the **connective tissue** that brings coherence to complex ecosystems, enabling various parts of the organization to work in harmony.

1.3.1: Grounding and Empowerment

Rather than advocating for rigid distinctions between roles, I propose **empowering architects with a data-informed, outcome-oriented mindset**. In this approach, architectural roles are **adaptable**, evolving to meet

the organization's needs while remaining aligned to deliver meaningful value.

Grounded Architecture refers to anchoring architectural work in the organization's reality:

- Grounded in **data** to inform decisions
- Grounded in **people** to foster collaboration and shared understanding
- Grounded in **pragmatic operating models** that guide action

To adopt this approach, you do not need to restructure your teams or redefine every role. What matters most is **how architects operate**—as collaborators, connectors, and enablers of aligned outcomes.

1.3.2: Adaptability Over Rigidity

IT architecture is **not one-size-fits-all**. Its form and focus should adapt to the context of each company. In some organizations, the emphasis may be on **business alignment**—ensuring that technology investments directly support the mission. In others, the focus may be on **technical depth**—working closely with engineers to build scalable, secure, high-performing systems.

Regardless of emphasis, the **core function remains the same**: to bring coherence to complexity and alignment to decisions. Therefore, I define “architect” not by job title, but by function—anyone who bridges the gap between **strategic intent and technical execution** is an architect.

This adaptable perspective allows organizations to maximize the benefits of architecture while enabling architects to adjust their contributions based on what is most needed—strategy, integration, governance, or mentorship.

1.3.3: Accepting the Challenges

Of course, this broader and more flexible view brings **new challenges**.

Unlike traditional frameworks, which have predefined roles, clear hand-offs, and rigid responsibilities, the Grounded Architecture approach **requires architects to adapt continuously**. It values relevance over formality and responsiveness over structure.

This approach can be demanding.

It requires architects to:

- Shift fluidly between **strategic thinking and hands-on guidance**
- Engage regularly with **diverse stakeholders**
- **Recalibrate their priorities** as business needs evolve
- Make decisions amid **ambiguity and uncertainty**

It also demands sound judgment—knowing **when to lead, when to advise, and when to step back**.

However, with these challenges come significant benefits.

Architects who embrace this approach are better equipped to respond to real-world needs. They help build solutions that are **context-aware, collaborative, and aligned with long-term goals**. They foster trust, promote innovation, and unlock value—not by controlling systems, but by guiding them with insight, empathy, and adaptability.

This book is a call to **reimagine architecture as a living, collaborative, and responsive practice**. One that moves beyond rigid frameworks and static organizational charts. One that values architectural success not in titles or artifacts, but in **outcomes, alignment, and impact**.

It may not always be the easiest path, but I believe it is the one most capable of delivering lasting value in complex, modern organizations.

1.4: Key Influences

The **Grounded Architecture** approach is shaped not only by my experiences but also by the work and ideas of others who have inspired and informed my thinking.

1.4.1: The Architecture Elevator — Gregor Hohpe

One of the most influential voices behind this work is **Gregor Hohpe** and his concept of the **Architecture Elevator**¹⁸. Gregor provides a vivid illustration of the modern architect as someone who navigates between the “penthouse,” where strategy is developed, and the “engine room,” where technology is built.

His perspective of architects as enablers of alignment, friction reduction, and transformation strategy has been foundational to my practice of Grounded Architecture. Much of what I share in this book reflects the valuable lessons learned from applying Gregor’s ideas in real organizational contexts.

1.4.2: Staff+ Engineering — Tanya Reilly & Will Larson

Another major influence has been the evolution of **Staff+ engineering** roles, which present a contemporary and practical view of technical leadership. Key texts such as:

- [The Staff Engineer’s Path](#) by Tanya Reilly¹⁹
- [Staff Engineer](#) by Will Larson²⁰

...offer a clear and relatable framework for understanding **what senior technical leadership entails outside of traditional management tracks**. These models have significantly shaped my thoughts on the responsibilities, mindset, and career development of modern architects.

¹⁸<https://architectelevator.com/>

¹⁹<https://www.oreilly.com/library/view/the-staff-engineers/9781098118723/>

²⁰<https://staffeng.com/guides/staff-archetypes/>

Staff+ engineers and architects often encounter similar challenges—balancing long-term vision with day-to-day decisions, influencing without authority, and collaborating across boundaries. The overlap between these roles is considerable, and these resources have informed my perspective on **architectural maturity, influence, and impact**.

1.4.3: Many More Sources

Grounded Architecture is built on a broad foundation of books, frameworks, articles, and tools—too numerous to list in one place. I have collected and curated many of these resources in the online **Bookshelf appendix**²¹, where you can explore what has influenced this work and perhaps find inspiration for your own journey.

²¹<https://grounded-architecture.io/bookshelf>

1.5: Why This Book?

This book serves as a way to **generalize and share years of experience** as a practicing architect and architecture leader. It originated from a need to **clarify my thinking, educate others, and create shareable resources** that can help individuals and organizations better understand and apply modern architectural practices.

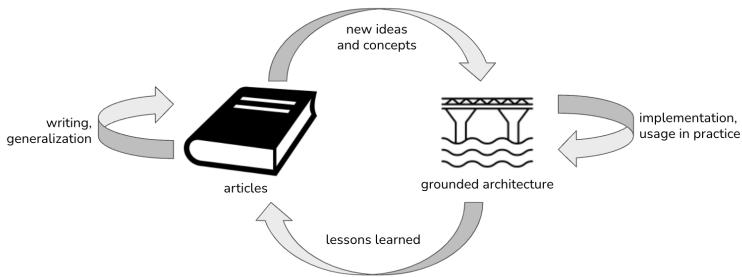


Figure 2: Writing helped me refine my thinking, evolve tools and principles, and share lessons learned.

1.5.1: Writing to Think

Writing forces clarity. As **Gregor Hohpe** once said, “*Every sentence you write frees up brain cells to learn new things.*” This was true for me as I began to shape my ideas into written form. Writing this book helped me:

- **Organize my thoughts**
- **Identify gaps and contradictions in my approach**
- **Enhance my frameworks and tools**
- **Discover new insights through reflection**

1.5.2: Writing to Teach

Another motivation was education—specifically, educating architects and technical leaders within the organizations I’ve worked for. I found that well-written, accessible material helps:

- Explain the “why” behind architecture
- Standardize terminology and mindset
- Create alignment across teams and levels

As Gregor Hohpe also points out, writing has several distinct advantages over spoken communication:

- **It scales** — You don’t need to be present to reach your audience.
- **It’s faster** — People read 2–3 times faster than they can listen.
- **It’s searchable and reusable** — Readers can revisit it, quote it, and reference it.

In short, **written content makes architectural thinking more portable and enduring**—ideal for spreading modern ideas across large, distributed organizations.

1.5.3: Writing to Share (and Learn More)

By documenting and refining my experiences, I hope to make this content **valuable for others**—whether they are facing similar challenges or are simply curious about evolving their architectural practice. I also look forward to:

- Learning from others’ feedback
- Comparing notes with peers
- Continuing the conversation about what modern IT architecture can become

This book is not a final answer; it’s a **living reflection** of one practitioner’s journey. My hope is that it sparks new insights, invites discussion, and helps architects—and those who work with them—develop more grounded, human, and effective practices.

1.6: A Part of the Bigger Picture: A Trilogy in Four Parts

This book is just one piece of a broader ecosystem of **open-source tools and resources** I have developed over the past decade to support my work as an architect. Together, they form a **practical toolkit** designed to make architecture more **data-informed, efficient, and impactful**.

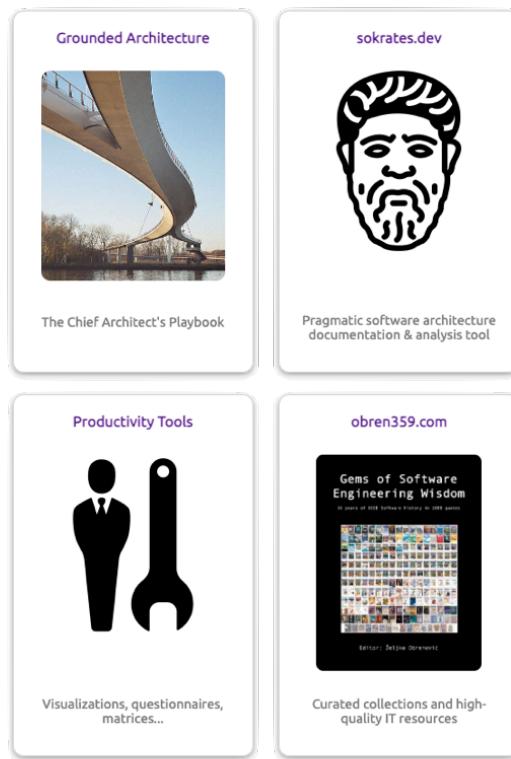


Figure 3: Grounded Architecture is part of a collection of open-source tools and learning resources developed over the past 10 years.

1.6.1: Other Tools in the Collection

- **Sokrates²²**

A polyglot source code analysis tool for extracting actionable insights from codebases.

- Understand your codebase through visualizations of **size, complexity, coupling, and team topologies**.
- Sokrates is designed to be **pragmatic, lightweight, and accessible**, making it perfect for architects who need fast, clear insights from actual repositories.

- **Productivity Tools²³**

A curated set of over **100 free online tools** I use regularly to support my daily architectural and engineering tasks.

- This collection includes everything from text parsers and visualization generators to JSON formatters and other architectural aids.

- **359° Overview of Tech Trends²⁴**

A hand-picked, constantly updated collection of **podcasts, videos, and talks** from over 20 high-quality sources, including IEEE, ACM, SE Radio, GOTO, and Martin Fowler.

- This resource is designed to help architects **learn quickly and stay current** in a noisy, fast-changing world.
- It includes **Ph.D. theses, conference talks, and trusted expert commentary** to support deep, strategic thinking.

For more about these tools and how I use them in practice, visit my homepage at **obren.io²⁵**.

Together, these resources support the vision behind *Grounded Architecture*, helping architects stay connected, curious, and grounded in both data and real-world needs.

²²<https://sokrates.dev>

²³<https://obren.io/tools>

²⁴<https://www.obren359.com/>

²⁵<https://obren.io/>

1.7: A Bit of Personal History

The ideas and practices presented in this book are based on **years of hands-on experience** across various roles, organizations, and industries. Each chapter of my career has contributed essential insights that form the foundation of the **Grounded Architecture** approach.

1.7.1: AVIV Group, eBay Classifieds, and Adevinta

Much of the work in this book stems from my current role as **Chief Architect at AVIV Group**, as well as my earlier positions as **Principal Architect at eBay Classifieds** and **Adevinta**. These experiences have enabled me to lead architectural practices in **large, complex, and rapidly evolving organizations**, skillfully balancing long-term strategy with everyday realities.

1.7.2: Software Improvement Group (SIG)

In the earlier stages of my career, I was a **consultant and analyst** at the **Software Improvement Group (SIG)**²⁶, where I discovered the power of **data-informed decision-making**. This role taught me how to extract meaningful insights from systems and how invaluable these insights are for guiding architectural choices.

Building on that experience, I developed **Sokrates**²⁷, an open-source tool designed to extract data from source code repositories to visualize **system complexity, team structures, and organizational dynamics**. This work significantly influenced the **Lightweight Architectural Analytics** pillar of the Grounded Architecture framework.

1.7.3: Incision

My tenure as CTO of **Incision**²⁸, a healthcare startup, provided a direct view of the challenges in **building and managing a technology organization** from scratch. It was an intense, insightful, and humbling journey

²⁶<https://www.softwareimprovementgroup.com/>

²⁷<https://sokrates.dev>

²⁸<https://incision.care>

that deepened my understanding of architecture at the intersection of **delivery, constraints, and leadership**.

1.7.4: Research in Design and Systems Thinking

Before entering the industry, I worked as a researcher at the **Dutch Center for Mathematics and Computer Science (CWI)**²⁹ and **Eindhoven University of Technology (TU/e)**³⁰. This academic background equipped me with skills in **rigorous data analysis, modeling, and research methods**.

One project from that period that significantly shaped my perspective was **Design Instability**³¹, a series of essays co-authored with Erik Stolterman. We explored the commonalities in architectural practices across disciplines, including classical design, UX design, and software engineering. This exploration helped me appreciate the **value of cross-disciplinary thinking**, a crucial skill for modern architects working at the intersection of technology, people, and business.

1.7.5: Developer Roots

Finally, I greatly value my early experience as a **hands-on software developer**. It's important for architects to remain close to the code and to the developers who write it. This technical grounding has proven essential in all my subsequent roles.

Together, these experiences have shaped a practice that is **data-informed, people-oriented, and grounded in real-world complexity**. This book reflects those lessons—not as a theoretical model, but as a **practitioner's field guide** for building architecture that truly works: **inside real organizations, with real people, solving real problems**.

²⁹<https://www.cwi.nl/en/>

³⁰<https://www.tue.nl/en/>

³¹<https://design-instability.com/>

1.8: Stay Connected

You can find additional resources online at:

- <https://grounded-architecture.io>³²

Feel free to follow me on LinkedIn to see what I am up to:

- <https://www.linkedin.com/in/zeljkoobrenovic>

³²<https://grounded-architecture.io/>

1.9: Acknowledgments

I sincerely thank all members of AVIV Group's Architecture Center of Excellence and the eBay Classifieds Virtual Architecture Team (VAT). Your thoughtful feedback, honest conversations, and challenging questions were crucial in shaping the ideas presented in this book. Our discussions were not only insightful but also essential.

I want to thank Peter Maas and Brent McLean for their leadership and vision. Your commitment to developing a data-informed architecture practice has laid the foundation for much of the work described here. Your support and encouragement have made a real difference.

This book reflects a shared journey, and I am grateful for everyone who has walked part of the path with me.

The cover image is a photo of Nesciobrug³³. Credit: the botster, CC BY-SA 2.0, via Wikimedia Commons.



image by henk monster cc by 3 0 via wikimedia commons

³³https://commons.wikimedia.org/wiki/File:Nesciobrug_4.jpg

2: Context: Fast-Moving Global Organizations



image by paul brennan from pixabay

IN THIS SECTION, YOU WILL: Understand the context in which the ideas in this book developed.

KEY POINTS:

- To better understand any idea or solution, it is crucial to understand the context in which this idea developed.
- The Grounded Architecture approach has evolved in the context of global, loosely coupled organizations that are diverse, with nonlinear growth dynamics, and under transformation pressures.

My work on creating and running an architecture practice isn't just a lofty idea; it's a **practical approach** sharpened from **real-world experience**. My perspective comes from lessons I learned as the Chief Architect at AVIV Group and the Principal Architect at eBay Classifieds and Adevinta.

To better grasp presented ideas or solutions, it is helpful to understand the problems we were trying to solve and **the context** in which these ideas were born. Here's a peek into the context that shaped my Grounded Architecture approach:

- **Global scale:** the organizations I worked in were operating across multiple countries and continents with millions of users.
- **Multidimensional diversity:** these organizations were diverse in terms of their customer base, workforce, business models, team topologies, and technology stacks.
- **Nonlinear growth dynamics:** in addition to organic growth, complex organizations change their portfolio through mergers and acquisitions of new businesses or divestments.
- **Synergies and transformation pressures:** complex organizations want to exploit the benefits of economies of scale and reduce duplication of efforts.
- **Decentralized, loosely coupled organizational units:** organizational units have significant autonomy while working together on common goals.

2.1: Global Scale

I have honed my approach within genuinely global and multicultural organizations on a massive scale:

- Operating across numerous **geographies, cultures, and languages**,
- Serving **millions of users** daily,
- Collaborating with thousands of software **developers** across hundreds of product and development **teams**,
- Implementing systems comprising hundreds of **millions of lines of source code**.



image by pete linforth from pixabay

Operating on a global scale introduces several compelling opportunities for organizations. It can significantly increase organizational effectiveness by **reducing duplication of effort** through centralized shared activities. Additionally, leveraging **economies of scale** allows for cost advantages, such as lowering the unit prices of utilized technologies. Global operations also enhance **business resilience and flexibility**, enabling compensation for local market fluctuations with global resources.

The expansive talent pool available to global organizations supports local and international initiatives. Moreover, these organizations possess significant resources to invest in supporting nonlinear growth through mergers and acquisitions (M&As).

However, the global and massive scale also presents numerous challenges. It results in **high organizational complexity**, with thousands of potential communication channels within the organization. The **complex technology landscape** entails numerous interconnected services. Managing a large talent pool incurs **high workforce costs**. Furthermore, such organizations face high computing resource expenses due to the need to serve a vast customer base around the clock. The operational complexity increases with high and variable customer demands across multiple locations. Additionally, global organizations have a **vast attack surface**, with many potential entry points for attackers. Lastly, any manual process, such as creating an organizational or technology landscape overview, is limited due to the scale involved.

Balancing opportunities and challenges on a global scale has been one of the most demanding and rewarding aspects of my architectural work. Such a magnitude makes any manual process inefficient and difficult to scale. The global scale was one of the main drivers behind the aggressive datafication of our an architecture practice. It has also led us to create more decentralized collaborative networks and operating models to execute and track decision-making across the board.

2.2: Multi-Dimensional Diversity

The organizations I worked with were incredibly diverse across multiple dimensions:

- **Cultures:** A varied workforce and clientele, both local and remote.
- **Organization:** Units of different sizes, complexities, and organizational styles.
- **Product:** Diverse product features catering to various markets and customer segments.
- **IT Architecture:** Combination of legacy systems and modern approaches.
- **Technology:** Numerous programming languages and thousands of third-party libraries, frameworks, and services.



image by simon from pixabay

For instance, I worked with organizational units differing in several aspects, including **unit size**, which ranged from hundreds of employees to just a dozen. The **team topologies** varied, spanning from single-team setups to hierarchical team organizations. Additionally, the **architectural roles** varied, with some units having dedicated local architecture teams and lead architects. In contrast, in smaller units, team members handled architectural duties alongside other responsibilities.

Similarly, technology-wise, we managed a range of styles in active production systems, from legacy **monolithic** applications to intricate modern **microservice** and **serverless** ecosystems. Each organizational segment had its own unique history and legacy systems. Our technology stack was extensive, covering multiple mainstream technologies. The infrastructure included several public cloud providers such as AWS, GCP, Azure, and custom-built private data centers. Our systems employ various application technologies, including database technologies like MySQL, PostgreSQL, MongoDB, Cassandra, AWS RDS, and more. The backend programming languages used were Java, C#, Go, Scala, PHP, Node.js. We used Swift, Objective-C, Java, Kotlin, Flutter/Dart, and more for mobile app programming. The frontend programming languages and frameworks included React, AngularJS, Vue, jQuery, and others.

Diversity offers several **opportunities**, including increased technology **innovation**. A diverse workforce can explore a variety of technologies and tools creatively. It also leads to better implementation because access to a broader pool of diverse resources allows for the selection of the best tool for the job.

However, diversity also brings challenges. One such challenge is **increased complexity**, resulting in a higher system landscape complexity and greater cognitive load for teams mastering numerous topics simultaneously. Additionally, there is **reduced flexibility**, as expertise spread across many domains and technologies limits reorganization possibilities. Furthermore, diversity can lead to higher **technical debt** due to multiple technology stacks, increasing legacy components, and outdated technologies. While diversity is a rich source of new possibilities from an architectural perspective, it always necessitates carefully managing complexity.

Diversity has influenced our an architecture practice in multiple ways. It has led to the development of lean tools that can cover a broad range of technology stacks rather than the adoption of specialized ones that can go

deep on one stack but cannot cover 95% of our landscape. We also adopted a more flexible governance model to help all teams in a practical way that is aligned with their diverse ways of working.

2.3: Nonlinear Growth Dynamics

Complex organizations like the ones I have worked in are often highly dynamic. These organizations frequently undergo significant growth, contraction, and reorganization, evolving both **organically** and **inorganically**.



image by pexels from pixabay

Organic growth refers to internal expansion driven by the company's own operations. **Inorganic change** involves acquiring other businesses, opening new locations, or divesting parts of the company.

Nonlinear growth, in particular, can be advantageous in several scenarios. It can **rapidly increase the customer base** or introduce new market segments. Additionally, such changes can **accelerate innovation** by incorporating new technologies or services.

However, nonlinear growth dynamics significantly impact architectural activities. The sudden integration of new companies **increases organizational complexity**, introducing many new units. Acquiring a new company also **brings in new technology and engineering units**, along with their unique processes and technology stacks. Furthermore, these nonlinear dynamics **necessitate a flexible architecture** to accommodate potential divestitures.

Nonlinear growth offers substantial benefits but also challenges managing increased complexity and the need to maintain architectural flexibility. In terms of its impact on an architecture practice, such dynamics lead to constantly high levels of complexity and more uncertainty. This has led us to prioritize the creation of better transparency to track changes that such dynamics introduce. We also needed to collaborate closely with business and finance stakeholders on developing tools for economics and risk modeling of investments and divestments.

2.4: Synergy and Transformation Pressures

Complex organizations aim to grow not just in size, but also in efficiency by leveraging economies of scale, cost synergies, and enhancing their capacity for innovation. Our investors expect us to become **more than the sum of our original parts.**



image by mustangjoe from pixabay

Pursuing synergies and transformations offers several opportunities, such as **cost reductions** through less duplication and lower expenses. **Accelerated innovation** can occur as savings from cost reductions free up resources for new developments. Additionally, creating synergistic components enables more possibilities for **reuse and sharing**, while well-executed transformations result in increased efficiency and **lower unit costs**.

However, striving for synergies and efficiency presents challenges. There is a need for significant **initial investment** to realize benefits, which

carries high risks. Performance pressure arises as teams must deliver excellent **short-term results** while undergoing significant transformations. Balancing transformation activities with regular work can temporarily **reduce productivity**. Moreover, post-transformation, the organization and technology landscape may become more complex due to increased dependencies, such as reusing central services.

The pressure to achieve synergies and efficiency can lead to high expectations and complicate regular architectural work. Nonetheless, these forces also create numerous opportunities for growth and improvement. For an architecture practice, these pressures created a strong need for better tracking project costs, value, and risks. Being able to calculate and back with data decisions for both innovative projects and legacy retirements was a critical aspect of our work.

2.5: Decentralization and Loose Coupling

Researcher Karl Weick developed the concepts of tight and loose coupling to describe organizational structures, initially in educational institutions and later applied to diverse businesses. According to Weick, a **tightly coupled organization** has mutually understood rules enforced by *inspection and feedback* systems. In such organizations, management can directly coordinate different departments' activities according to a central strategy.

In contrast, a **loosely coupled organization** lacks some elements of a tightly coupled one. Employees have **more autonomy**, and different departments may operate with **little coordination**.



image by andrii yalanskyi from istock

Most organizational units I worked with were loosely coupled. Our companies frequently grew through acquisitions of companies in different marketplaces. Business strategies also promoted the independent evolution of local units to address local market needs more effectively and quickly. These units often enjoyed a high level of autonomy, frequently with their development teams and sometimes with local CFOs, CMOs, or CEOs.

Loose coupling offers several advantages. It provides **higher flexibility**,

allowing units to develop independently and address specific needs without synchronizing with other units. This flexibility leads to **reduced time-to-market**, as fewer dependencies enable marketplaces to rapidly change and evolve their products for local needs. Additionally, loose coupling **fosters innovation** by offering opportunities to quickly explore ideas in smaller contexts.

However, loose coupling also presents several challenges. It can lead to duplication of effort, as local market needs might differ but often have significant overlap in product features and technology, resulting in **redundant efforts** as each marketplace creates solutions for the same problems. This approach also increases **accidental diversity**, where limited synchronization may result in significantly different design and technology choices for the same problem, making it difficult to consolidate solutions, move personnel between teams, or benefit from economies of scale. Moreover, loose coupling results in **limited control**, as fewer dependencies and varying goals make it more challenging to implement changes across the organization.

From an architectural perspective, loose coupling presents an interesting challenge, often leading to a conflict between global alignment and control and local autonomy. For our architecture practice, decentralization and loose coupling led to many changes. We emphasized “hands off, eyes on,” leaving teams autonomy in their work but creating complete transparency based on data. Our operating model has a high level of decentralization to enable both scaling and alignment of architecture work.

2.6: Questions to Consider

To better understand any idea or solution, it is crucial to understand the context in which these ideas developed. When using ideas from this book, ask yourself how your organizational context differs from mine:

- *What are the unique characteristics of your organizational context?*
- *What is the scale of your organization? How it affects your architecture practice?*
- *How diverse is your organization?*
- *What are the growth dynamics of your organization?*
- *Are you experiencing synergy and transformation pressures?*
- *How (de)centralized is your organization?*

3: Goals: Adapting, Growing, and Using Data



image by bluehouse skis from pixabay

IN THIS SECTION, YOU WILL: Understand the requirements I identified for an architecture practice in complex organizations.

KEY POINTS:

- I identified the following needs that an architecture practice should support: Executing At Scale, Adaptivity, Improving the Quality of Decision-Making with Data, and Maximizing Organizational Alignment & Learning.

Grounded Architecture emerged as a necessity in response to our **intricate and multifaceted challenges**. The Grounded Architecture framework was designed to address these specific challenges. By moving away from manual processes and embracing automation, data-driven decision-making, and adaptive frameworks, we aimed to create a **more resilient and effective** an architecture practice.

In following sections I will outline a breakdown of the goals I set for an architecture practice:

- Goal 1: Executing At Scale,
- Goal 2: Adaptivity,
- Goal 3: Enhancing Decision-Making Quality with Data,
- Goal 4: Maximizing Organizational Alignment,
- Goal 5: Maximizing Organizational Learning.

3.1: Goal 1: Executing At Scale

Our organizations were like a bustling city with hundreds of teams and thousands of projects, each with its own unique complexity and requirements. Traditional, one-size-fits-all approaches to an architecture practice simply couldn't keep up with this dynamic environment. We needed a system that could support this vast and varied ecosystem. Grounded Architecture was designed to **handle such diversity at scale**, ensuring that teams and projects received the tailored support they needed without being bogged down by rigid processes.

Some of the success criteria for this goal included:

- Always having the **full transparency** about the technology landscape. Without full transparency, it isn't easy to understand the landscape's complexity or your work's context. This transparency should include good data and visualization of the size and quality of all source code repositories, public cloud accounts and technologies, private data centers, development efforts, etc.
- Having mechanisms and spaces to maintain **practical working relationships** with all development teams and key stakeholders. Knowing the organizational landscape and having spaces for engagement is crucial.
- Being able to scale and grow the organization **without** introducing significant **slowing of decision-making**. Finding the right balance between teams' autonomy and alignment is essential.

3.2: Goal 2: Adaptivity

In our dynamic environments, significant change is not just frequent; it's expected. Whether these changes are organic, like evolving business needs, or inorganic, like mergers and acquisitions, our architecture must be able to adapt swiftly. Grounded Architecture was crafted to be flexible and responsive, allowing us to pivot quickly in response to new challenges and opportunities. This adaptability ensures that our architectural framework remains relevant and effective, no matter how the organizational landscape shifts.

Some of the success criteria for this goal included:

- Being able to track and support **legacy and new technologies**, adapting this support as the organization transforms, grows, acquires new legacy, and adopts new technologies.
- Having **readily available data** for analyses of different business scenarios (e.g., retire legacy vs. investment in legacy, buy-or-build, divestments).
- Being able to **routinely onboard** and quickly understand the technology landscape of acquired companies.

3.3: Goal 3: Enhancing Decision-Making Quality with Data

Relying on gut feelings or individual opinions is always insufficient and risky when dealing with operations at scale. Decisions need to be based on solid data to **ensure accuracy and reliability**. Grounded Architecture aims to incorporate tools and mechanisms to support data-driven decision-making. By leveraging data and analytics, we can move away from subjective opinions and towards more objective, evidence-based decisions. This approach should enhance the quality of our decisions and facilitate their consistency and alignment with our organizational goals.

Some of the success criteria for this goal included:

- Always having **complete, up-to-date data** about all key elements of the organizational technology landscape.
- Having technical data connected with **product** and **business data** (e.g., vibrancy vs. public cloud costs).
- Making data available via **self-service tools** for the organization so that more people can make data-informed decisions.
- Ensuring **routine usage** of the data in decision-making.

3.4: Goal 4: Maximizing Organizational Alignment

In a global, fast-moving organization, misalignment can quickly become the norm. Different teams and departments might pursue conflicting objectives, leading to inefficiencies and confusion. Grounded Architecture aims to **serve as a cohesive force**, promoting alignment across the entire organization. Providing a clear, unified framework helps to minimize misalignments. It facilitates all parts of the organization working towards common goals. This alignment is crucial for maintaining efficiency and avoiding the chaos that can arise from disparate efforts.

Some of the success criteria for this goal included:

- Having pragmatic **standardized guidelines** and best practices (e.g., golden paths) for technology use, ensuring consistency across the organization.
- Fostering a **culture of collaboration** and knowledge sharing among teams to align on technology choices and implementation strategies.
- Regularly **reviewing and updating** technology standards to align with evolving business needs and industry trends.
- Identifying and eliminating **redundant processes** and activities to streamline operations and reduce wasted resources.

3.5: Goal 5: Maximizing Organizational Learning

Staying current with emerging technologies and industry trends is essential for maintaining a competitive edge. Still, it can be challenging when dealing with the demands of legacy systems. Grounded Architecture should facilitate continuous learning and growth. It supports the rapid adoption of new technologies and encourages **ongoing education and training**. Grounded Architecture should ensure we always have the best tools and knowledge to drive innovation and improvement.

Some of the success criteria for this goal included:

- Organizing frequent workshops, seminars, and training sessions to facilitate **sharing knowledge** and best practices across the organization.
- Encouraging employees at all levels to contribute to and participate in knowledge-sharing initiatives, promoting a **culture of continuous learning**.
- Developing platforms and tools that enable **easy access** to shared knowledge and resources, enhancing collective expertise.
- Involving **diverse stakeholders**, including developers, managers, and end-users, in discussions and decision-making processes related to architecture and technology.
- **Creating opportunities** for employees from various departments to engage in architectural planning and feedback sessions to learn from each other.

These criteria aim to create an inclusive and dynamic environment where knowledge is freely shared, and diverse contributions are not just welcomed, but valued.

3.6: Questions to Consider

Knowing what goals an architecture practice needs to support in your organization is crucial to defining structures and measuring your impact. Some of the plans may be universally applicable. Others may be unique to your context. Ask yourself the following questions:

- *What is the scale of your an architecture practice? Does your scale require special measures to ensure your an architecture practice efficient operations?*
- *What are the key decisions you need to make? Do you have the data to base your decisions?*
- *How aligned are units in your organizations? How much friction is there? How can an architecture practice help?*
- *How much is your organization learning? How is the learning supported?*
- *How stable is your organization? How likely is it that significant changes will occur in your organization?*

Part I: Grounded Architecture Framework

4: Grounded Architecture Framework: Foundations

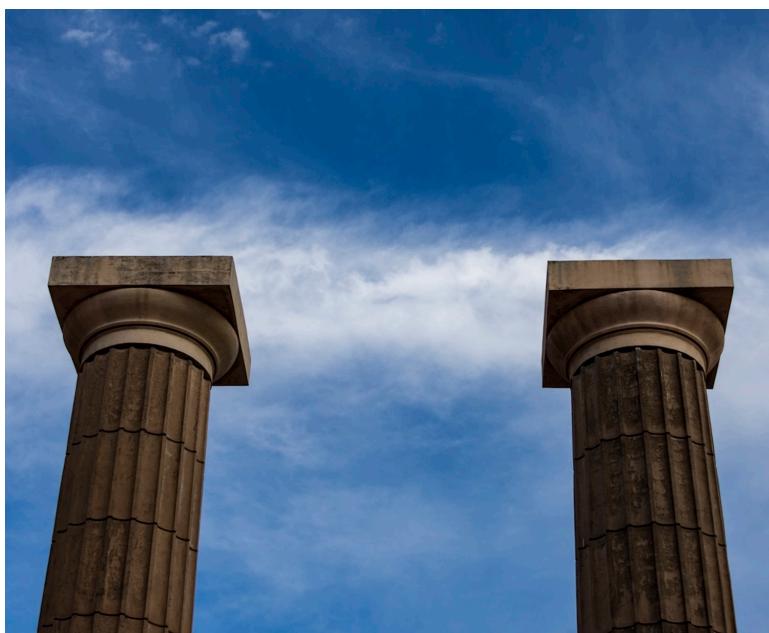


image by dario truco from istock

IN THIS SECTION, YOU WILL: Get an overview of the Grounded Architecture framework: Lightweight Architectural Analytics, Collaborative Networks, and Operating Model.

KEY POINTS:

- I introduce three elements of Grounded Architecture Framework: Lightweight Architectural Analytics, Collaborative Networks, and The Operating Model as an approach to setting organizational structures for a modern IT architecture practice.
- Prioritizing people interactions and data over processes and tools, Grounded Architecture aims to connect an architecture practice to all organizational levels as an antidote to the “ivory tower” architecture.

In this section of the book, I will introduce the **Grounded Architecture** framework—my practical approach to establishing an effective and scalable architecture practice within complex organizations.

I chose the name *Grounded Architecture* intentionally. It highlights the need to avoid creating an “ivory tower” architecture practice—one that is disconnected from the daily realities of the business. In a **fast-moving, global, and diverse environment**, such disconnection is not only inefficient but also dangerous.

The Grounded Architecture framework is designed as a **pragmatic and adaptive solution**. Its aim is to keep architecture deeply rooted in the organization, prioritizing **people’s interactions** and **real-time data** over rigid processes and cumbersome tools. The goal is to **embed architecture across all levels and parts of the organization**, serving as an antidote to traditional, top-down approaches that often fail to make a meaningful impact.



Figure 1: The Grounded Architecture framework: the foundations.

The Grounded Architecture framework consists of three core elements (see Figure 1):

- Lightweight Architectural Analytics (Data)
- Collaborative Networks (People)
- The Operating Model (*which will be covered in a later section*)

4.0.1: Lightweight Architectural Analytics

This part is a **system of tools and resources** that provides architects with a **real-time, curated view** of the organization's technology landscape. It enables **data-informed decision-making** at every level.

For more information, refer to the [Lightweight Architectural Analytics section](#).

4.0.2: Collaborative Networks

These networks **connect everyone involved in architecture throughout the organization**—from central teams to embedded architects and tech leads. They are essential for ensuring that architectural decisions are **relevant, actionable, and aligned** with actual needs.

See the [Collaborative Networks section](#) for more details.

4.0.3: The Operating Model

The [Operating Model](#) ties everything together. It introduces **structures, routines, and roles** that connect people, data, and decisions into a **cohesive, collaborative, and impact-driven practice**. We will explore this in a dedicated section later in the book.

Now that we have completed a high-level overview of the Grounded Architecture framework, let's dive deeper into each of its elements and see how they come to life in practice.

5: Lightweight Architectural Analytics



image by ko_orn from istock

IN THIS SECTION, YOU WILL: Understand how to use diverse data sources to support architecture decision-making processes and get concrete tips on creating architecture-centric data tools.

KEY POINTS:

- Lightweight Architectural Analytics serves as a medium to create a complete, up-to-date picture of critical elements of the organization's technology landscapes.
- Such analytics provides an architecture-centric view of data about a technology landscape based on source code analyses, public cloud billing reports, vibrancy reports, or incident tickets.
- To facilitate the creation of Lightweight Architectural Analytics, I have been creating open-source tools that can help you obtain valuable architectural insights from data sources, such as source code repositories. Check out open-source architecture dashboard examples¹ and Sokrates².

"If we have data, let's look at data. If all we have are opinions, let's go with mine." — Jim Barksdale

In every organization where I have built an architecture practice, I have strongly—some might say obsessively—emphasized the importance of data. One of the first steps I take is to establish a **Lightweight Architectural Analytics** capability (Figure 1). This step is crucial for obtaining a **comprehensive and up-to-date view of the organization's technology landscape**.

¹<https://zeljkoobrenovic.github.io/grounded-architecture-dashboard-examples/>

²<https://sokrates.dev>

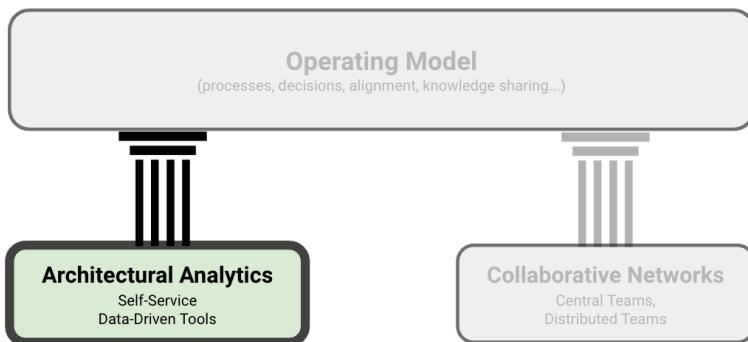


Figure 1: Grounded Architecture Framework – Lightweight Architectural Analytics

5.1: Why I Start With Data:

Manual documentation simply does not scale. It becomes outdated quickly and requires excessive effort to maintain. In contrast, **data is scalable, repeatable, and reliable**. When executed correctly, data serves as the foundation for effective and consistent architectural decision-making.

5.1.1: The Good News: You Already Have the Data

The good news? **Most large organizations already possess ample data**. It is just scattered across various tools, teams, and formats.

With the right combination of:

- **Automation** (to collect and maintain data)
- **Curation** (to clean and contextualize the data),

...you can unlock valuable architectural insights far more easily than you might expect.

5.1.2: Why “Lightweight”?

I use the term **lightweight** intentionally. This approach does not involve purchasing expensive software or establishing a massive data warehouse from day one. Instead, it is about:

- Starting small and simple
- Using open-source or low-cost tools
- Focusing on what's actionable rather than just what's impressive

You can derive real value using a handful of smart scripts, well-designed dashboards, and a willingness to experiment. The goal is to obtain just enough structure and visibility to facilitate better decisions—without creating a new bureaucracy.

This section will explore how Lightweight Architectural Analytics works in practice:

- What data is useful
- How to collect it
- How to visualize and share it
- And how to use it to drive alignment, reduce waste, and support effective architecture at scale

If architecture involves guiding complex systems through change, then data is the **compass** that helps keep us oriented in the right direction.

5.2: Examples of Lightweight Architectural Analytics Tools

To clarify what I mean by **Lightweight Architectural Analytics**, I will share some concrete examples from my recent work. These tools are part of a modular dashboard we have developed and extensively used at AVIV Group.

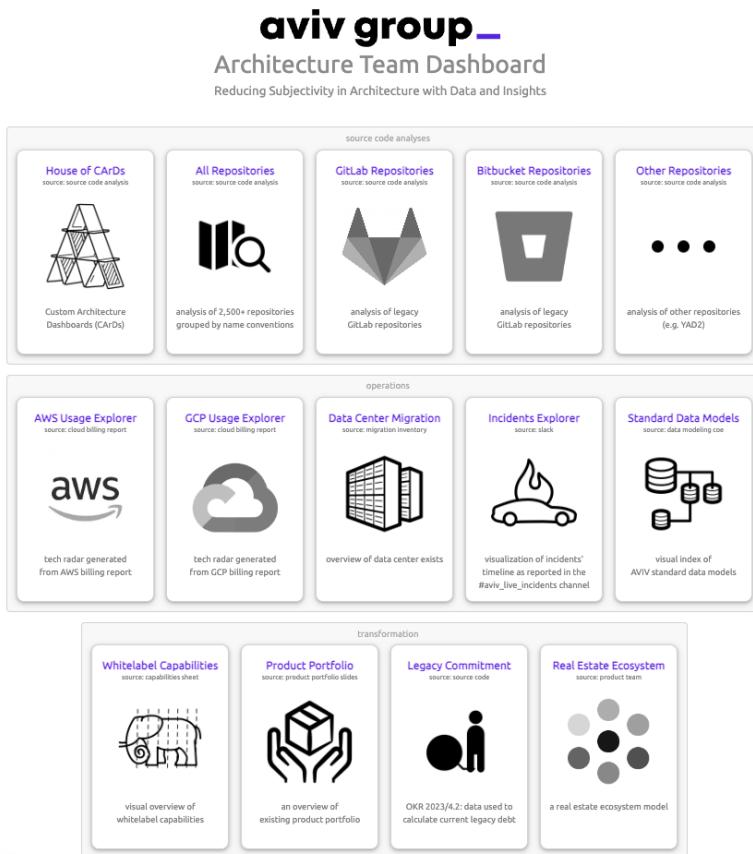


Figure 2: Start page of the architecture analytics dashboard used at AVIV Group.

I typically implement **Lightweight Architectural Analytics** as a **collection of focused data applications aggregated into a simple dashboard**. These tools are built on top of **existing organizational data** and provide accessible insights with minimal setup and maintenance.

Each application draws from one or more of the following **data streams**, which are often readily available in large organizations:

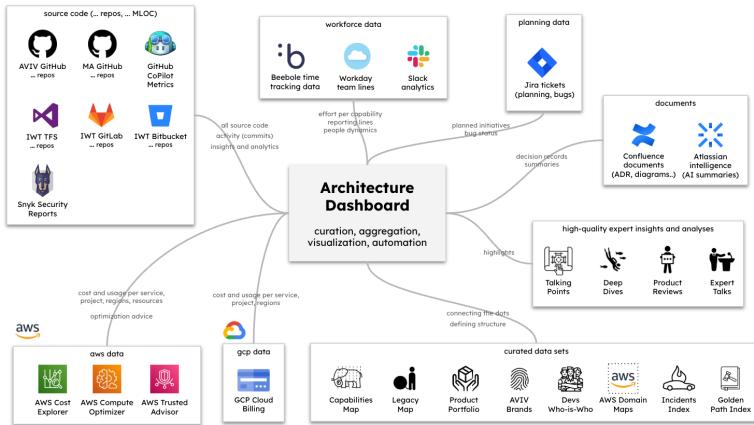


Figure 3: Overview of core data sources powering the architecture dashboard.

Source Code Repositories: Your source code is a valuable source of architectural signals. By analyzing:

- Commit histories
- Complexity trends
- Contributor patterns

...you can uncover valuable insights into **technical debt**, **team interactions**, and **system hotspots**.

Public Cloud Billing Reports: Cloud billing data reveals:

- Which services are being utilized
- Areas of potential inefficiency

- Cost trends across **regions, teams, and services**

This information supports more accountable cloud usage and budget-conscious design decisions.

Incident Reports: Incident data highlights:

- Recurring issues
- Weak points in system design
- Dependencies between systems and teams

Analyzing these trends can improve **reliability, resilience, and incident response planning**.

Business & Product Metrics (e.g., User Activity): By tracking core usage metrics, such as **user activity** or **system vibrancy**, architects gain visibility into how systems support real-world business outcomes, providing an essential link between technology and products.

Messaging & Collaboration Tools: Platforms like Slack offer insights into:

- Team collaboration patterns
- Frequent technical discussions
- Emerging pain points or themes

These insights help architects stay connected to the operational realities without the need for constant meetings.

In the following sections, I will guide you through a selection of these **data-driven architecture tools**, explaining how they were built, the insights they offer, and how they have facilitated architectural decision-making at scale.

5.2.1: Example 1: Source Code and Commit History Analytics

Your **source code** and **commit history** are a valuable source of insights—an often-overlooked asset in architectural work. Within that history lie powerful signals about your **technology stack, team dynamics, dependencies, and the quality and structure of your systems**.

5.2.1.1: Meet Sokrates: An Architect's X-Ray Vision

To help you harness this potential without becoming overwhelmed, I developed and actively maintain an open-source tool called **Sokrates**³.

Sokrates is a free, lightweight tool that analyzes codebases across multiple repositories and generates intuitive, visual reports that are:

- **Easy to explore**, even for non-developers.
- **High-level and strategic**, ideal for CTOs and architecture reviews.
- **Detailed and tactical**, useful for in-depth analyses and code critiques.

It allows you to **zoom out** for a broad overview of your organization's code and **zoom in** to inspect specific teams, technologies, or problem areas. Think of it as **an architect's x-ray vision** into your source code ecosystem.

³<https://sokrates.dev>

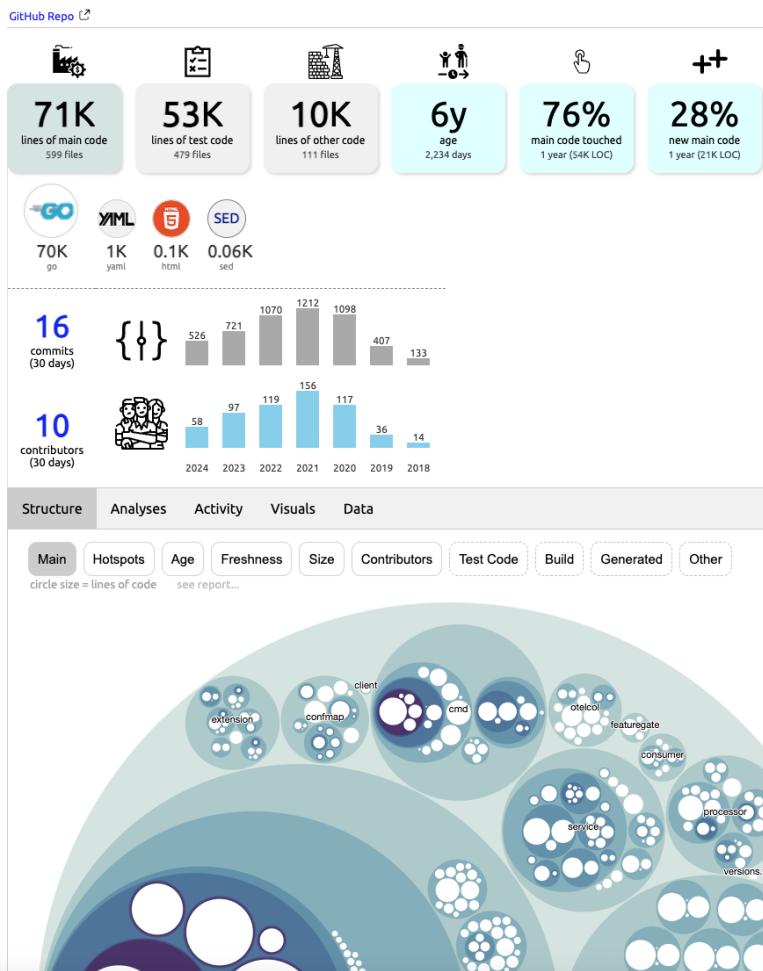


Figure 4: Screenshot from a Sokrates report dashboard.

5.2.1.2: Show Me, Don't Just Tell Me

Curious what this looks like on a larger scale? Here are a few real examples of **Sokrates in action**, analyzing large open-source landscapes:

- **Apache Software Foundation**⁴
 - 1,000+ repos · 180M+ lines of code · 22,000+ contributors
- **Facebook / Meta**⁵
 - 800+ repos · 120M+ lines of code · 20,000+ contributors
- **Microsoft OSS**⁶
 - 2,400+ repos · 100M+ lines of code · 18,000+ contributors
- **Google OSS**⁷
 - 1,600+ repos · 200M+ lines of code · 27,000+ contributors
- **Linux**⁸
 - 178 folders · 23M+ lines of code · 17,000+ contributors
- **Amazon OSS**⁹
 - 2,700+ repos · 130M+ lines of code · 13,000+ contributors

These reports illustrate how Sokrates can uncover patterns across thousands of projects—providing valuable insights into code health, ownership, and evolution.

5.2.1.3: Specialized Analyses for Architecture Insights

In addition to standard commit and repository analysis, I've developed several **specialized tools** that extract architecture-relevant signals from code and tooling:

- **CI/CD Insights** (via Travis & Jenkins Analyzers): Understand how teams build, test, and deploy.
- **Dockerfile Scanner**: Create a real-time **tech radar** of the runtime technologies used across teams.
- **GitHub PR Activity Monitor**: Measure deployment frequency and identify bottlenecks or silos.

⁴https://d3axxy9bcycpv7.cloudfront.net/asf/_sokrates_landscape/index.html

⁵https://d3axxy9bcycpv7.cloudfront.net/meta/_sokrates_landscape/index.html

⁶https://d3axxy9bcycpv7.cloudfront.net/microsoft/_sokrates_landscape/index.html

⁷https://d3axxy9bcycpv7.cloudfront.net/google/_sokrates_landscape/index.html

⁸https://d3axxy9bcycpv7.cloudfront.net/asf/_sokrates_landscape/index.html

⁹https://d3axxy9bcycpv7.cloudfront.net/amzn/_sokrates_landscape/index.html

5.2.1.4: Build Your Own

Sokrates is open-source and ready to use, but even if you choose to create your own tools or pipelines, I encourage you to:

- Experiment with **what matters most** to your organization.
- Connect **code signals to business context**.
- Share insights **visually** to maximize their impact.

Start small, automate early, and let the data guide you to architect smarter solutions.

5.2.2: Example 2: Public Cloud Usage Analytics

One of the key advantages of using public cloud platforms is the **built-in visibility and standardization** they provide. With **uniform automation and monitoring**, public cloud environments offer a level of **transparency** that is challenging to replicate in traditional infrastructures.

Providers like **Amazon Web Services (AWS)**¹⁰, **Google Cloud Platform (GCP)**¹¹, and **Microsoft Azure**¹² make **detailed usage data readily available**. This includes insights into:

- Which services are used and by whom
- What resource types and regions are consuming the most
- Team or project-level billing and budget trends
- Access permissions and ownership breakdowns

This data is invaluable for architects looking to **understand usage patterns**, **manage costs**, and **enhance architectural efficiency** across the organization.

To capitalize on this, I developed several open-source custom tools called **Cloud Usage Explorer**, which visualizes data from standard **cloud usage reports**. It transforms raw billing data into **clear, actionable dashboards** that track consumption, trends, and anomalies in near real-time.

¹⁰<https://aws.amazon.com>

¹¹<https://cloud.google.com/>

¹²<https://azure.microsoft.com/>

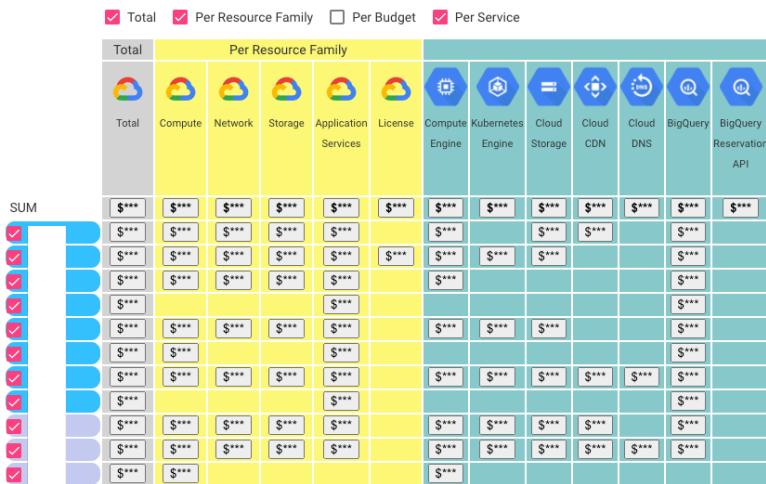


Figure 5: An example of a cloud usage explorer.

By leveraging this type of cloud usage data through Lightweight Architectural Analytics, you can:

- Identify underutilized or misconfigured resources
- Encourage cost-aware decision-making among teams
- Promote more efficient architectural design patterns
- Track the evolution of platform usage over time
- Detect siloed or duplicated efforts across teams

Importantly, you can accomplish all of this using data you already have, through **simple automation and smart visualization**—there's no need for enterprise-scale tooling right from the start.

5.2.3: Example 3: Business & Finance Data — A Hidden Gem

Finance departments are often the unsung heroes of enterprise data. Consider them the **Sherlock Holmes of the business**—relentlessly data-driven, meticulous, and always equipped with high-quality, structured information. While they are primarily known for tracking **costs, budgets,**

and forecasts, they often monitor much more than just these “dry” figures.

In my experience, finance teams frequently track **vibrancy**, **usage levels**, and **system engagement metrics**. They do this not out of mere curiosity, but to **connect financial performance with system usage**. This type of data is a **goldmine for architects**.

5.2.3.1: Why It Matters

By linking **financial data** (such as cloud costs) with **usage and vibrancy metrics**, you can:

- Uncover **underutilized systems** that are still incurring high costs.
- Identify platforms that have a **high business impact** and warrant further investment.
- Highlight **inefficiencies** in scaling, infrastructure, or usage patterns.
- Strengthen business cases for **optimization or redesign**.

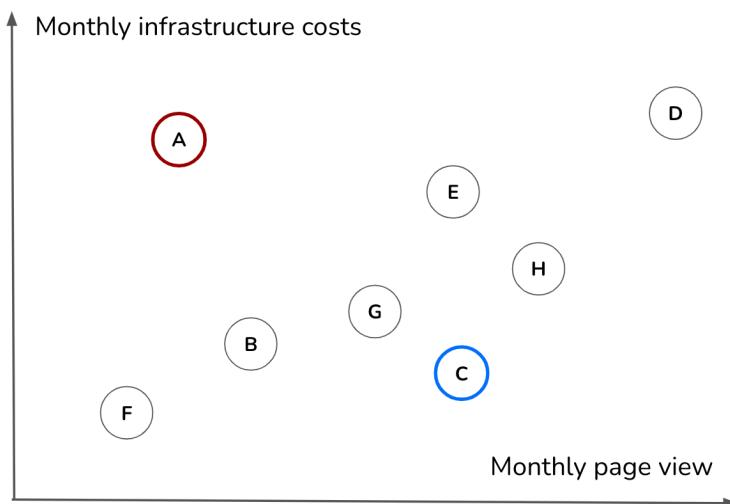


Figure 6: Combining data from different sources (e.g., cloud billing reports and vibrancy or revenue can lead to new insights (e.g., identifying inefficiencies in the application portfolio).

5.2.3.2: A Real-World Tip

When you're deep into architectural planning—whether mapping platforms, defining investment strategies, or discussing redesigns—**don't forget to consult the finance team.** They likely possess insights that can enhance your strategy and **anchor your architecture in real business value.**

While finance may not be present in architecture meetings, they hold some of the **most actionable data in the organization.** Integrating this data into your architecture discussions will enable you to make decisions that are not only technically sound but also economically and strategically aligned.

5.2.4: Example 4: Leveraging Generative AI

Once you've established a solid foundation of **curated, up-to-date, and accessible data**, you unlock a powerful new layer of value through **Generative AI tools**—such as ChatGPT, Gemini, or various **open-source LLMs.** These models serve as **intelligent companions** for architectural work, transforming your data into **interactive, dynamic conversations.**

Instead of relying solely on static dashboards or one-time reports, you can now ask natural language questions, such as:

- “Which teams are contributing the most to legacy systems?”
- “Which services have the worst cost-to-vibrancy ratio?”
- “Can you cluster microservices by team ownership and code churn?”
- “What anomalies occurred in our incident patterns over the past quarter?”

5.2.4.1: Why It Works

This approach is effective only when your underlying data is well-structured and readily available, which is precisely what **Lightweight Architectural Analytics** is designed to provide. Without that foundation, Generative AI lacks meaningful data to work with.

5.2.4.2: Practical Use Cases

Here are several ways I've utilized Generative AI tools in conjunction with architectural data:

Source Code Insights: Upload Sokrates reports or raw commit logs and prompt the AI with questions about:

- Code complexity hotspots
- Ownership overlaps or silos
- Changes in deployment frequency over time

Cloud Usage & Cost Optimization: Feed in cloud billing reports and ask the AI to:

- Detect underutilized services
- Highlight high-cost/low-impact areas
- Summarize team-specific cloud trends

Incident Pattern Detection: Analyze incident ticket data to:

- Identify frequent root causes
- Recognize recurring impacted services
- Discover trends in team-specific incident ownership

Business Metrics Correlation: Correlate system vibrancy, uptime, or adoption with:

- Team sizes
- Architectural changes
- Operational costs

Communication & Collaboration Trends: Use logs from tools like Slack to:

- Uncover dominant discussion themes
- Track inter-team collaboration patterns
- Identify shifts in focus over time

In each case, Generative AI serves as a **first-pass analyst**, revealing patterns, anomalies, or new ways of framing problems. This enables architects to **move faster, dig deeper, and think broader**.

5.2.4.3: Prompts as Reusable Thinking Tools

Prompts aren't merely one-time commands; they are a **Reusable, transparent expression of architectural intent**. Over time, prompts can become:

- A form of **lightweight documentation** for common questions and analyses
- A **repeatable template** for querying new datasets
- A method to **share architectural reasoning** across teams
- A **teaching tool** to assist junior team members facing real-world architectural challenges

Consider prompts as **scripts for architectural sensemaking**—fast, flexible, and easy to refine as your practice evolves.

For examples, please refer to the [Appendix on Generative AI Prompts¹³](#), where I've included ready-to-use prompt templates and sample datasets to help you get started.

Generative AI doesn't replace architectural judgment; it **amplifies** it. When paired with good data and clear thinking, it becomes a powerful ally in addressing complexity, accelerating insight, and enhancing your practice.

¹³[gen-ai-prompts](#)

5.3: Requirements for Lightweight Architectural Analytics

Lightweight Architectural Analytics should serve as a central, trusted resource—your organization’s **source of truth** for architectural insights. It’s not merely a storage bin for random metrics; it’s a curated space of **authoritative, relevant, and accessible data** that enables informed decisions and confident actions.

Technically, you can start small. I’ve seen organizations build early versions using **Google Drive**, **Confluence**, or even **documents pinned to a wall**. However, I strongly recommend investing in a **better user experience and infrastructure**. When executed well, Lightweight Architectural Analytics becomes a **valuable organizational asset**—not a digital junk drawer.

Collecting data isn’t enough. To make your analytics useful, it must meet several key requirements.

5.3.1: Single Point of Truth

People need to know exactly where to go. Your analytics hub should serve as the **central reference** for all relevant architectural data—no guessing, no scattered files, and no contradictory versions. Ask yourself: *If someone starts tomorrow, would they be able to find what they need in one place?*

5.3.2: Curated for Quality

Data without trust is simply noise. You must take ownership of **curation**, ensuring that the data is accurate, meaningful, and transparent. Whenever possible, link back to **original data sources** so others can verify the facts. Think of yourself more as an editor than just a collector.

5.3.3: Curated for Usability

People need focus, not friction. Filter out irrelevant information. Highlight what matters. Design your tools and dashboards with **clarity** in

mind—investing in **user experience (UX)** isn’t a luxury; it’s essential for making the data usable. It’s not about having all the data; it’s about having the **right data, clearly presented**.

5.3.4: Kept Up to Date

Stale data is dangerous. Your system should refresh data **automatically** or through **repeatable, lightweight processes**. When people see that the data is fresh, they’ll be more likely to trust and use it.

5.3.5: Accessible to the Whole Organization

Architectural data shouldn’t be restricted to a select few. When you give teams access to insights that were once reserved for “architects” or “leadership,” you empower them to act more quickly, make better decisions, and reduce dependency. Transparency accelerates performance.

5.3.6: Used in Decision-Making

Curated data that sits unused is a wasted effort. Your analytics must be incorporated into **actual decision-making processes**—in design reviews, strategy sessions, prioritization meetings, and product planning. Good architecture is informed by data.

5.3.7: Built Like a Map

The best metaphor for this type of analytics is a **map**. A good map provides **orientation** in a complex landscape. It shows where the treasure is and warns you of where dangers lie. Your architectural data should do the same—helping people understand:

- What exists
- How things are connected
- Where things are healthy or broken
- Who is responsible
- What is at risk

- What should be done next

Like real maps, it should have **layers**—views that reveal **different dimensions** of your systems: performance, quality, dependencies, cost, ownership, activity, and more.

By meeting these requirements, **Lightweight Architectural Analytics** becomes more than just a tool; it becomes a **strategic capability**—something that makes your entire organization smarter, faster, and more aligned.

5.4: Tips for Building Lightweight Architectural Analytics

Every organization has its own quirks when it comes to data, but after establishing architectural practices across several companies, I've noticed some common patterns and practical strategies. If you're starting (or rebooting) a **Lightweight Architectural Analytics** initiative, here are some key tips to make your journey smoother—and perhaps even enjoyable:

5.4.1: Start With the Source Code

My motto is: "*Talk is expensive. Show me the code.*"

Why? Because **code never lies**. It's the most honest and up-to-date documentation of what's really happening in your systems. While people may forget details or creatively reinterpret them, your repositories will tell the truth—messy or not.

Modern IT systems store nearly everything in code: infrastructure, pipelines, configurations, and documentation. This makes source code the **richest and most reliable source of architectural insight**.

I use tools like **Sokrates**¹⁴ to scan repositories early on. You'd be surprised how often these scans reveal that a “simple legacy service” is, in fact, a spaghetti monster in disguise.

5.4.2: Connect With Finance and Governance

Another motto: "*Follow the money!*"

Finance and governance teams often collect **high-quality, structured, and underutilized data**—from cloud billing reports to technology cost breakdowns. This data is already being tracked for compliance and budget forecasting, so why not utilize it for architecture?

You don't need revenue projections or sensitive figures. Just access to usage and cost data can unlock **unexpected insights** about system efficiency, platform sprawl, and ROI blind spots.

¹⁴<https://sokrates.dev>

5.4.3: Maintain a Culture of Transparency

Transparency isn't just a nice-to-have; it's a strategic enabler.

A transparent organization shares information freely, trusts its people with data, and reduces red tape. When you eliminate the need for complex access control mechanisms or bureaucratic workflows, your architecture practice can move faster and remain better aligned with reality.

Transparency also **fosters trust**—people are more likely to use and contribute to your analytics when they see it is open and well-intentioned.

5.4.4: Own the Curation

Raw data doesn't generate value. **Curation does.**

You need to thoroughly understand your data sources, filter out the noise, and present the **most relevant and trustworthy insights** in a clear and consistent manner. This isn't just a backend task; it's a **UX design problem**.

As the curator of your Lightweight Architectural Analytics, think of yourself as:

- An editor, not just an aggregator
- A guide, not just a dashboard builder
- A storyteller, helping teams navigate their complex systems with clarity

5.4.5: Use Simple, Maintainable Infrastructure

Keep it light. Keep it lean.

I publish most of our analytics (like Sokrates reports) as **static resources hosted on GitHub Pages**. This approach eliminates the need for a backend, databases, and maintenance overhead.

Check out the [Architecture Dashboard Examples repository](#)¹⁵, which includes:

¹⁵<https://github.com/zeljkoobrenovic/grounded-architecture-dashboard-examples>

- The dashboard source code (HTML + JSON)
- Lightweight visualizations
- A live demo hosted [here](#)¹⁶

This design is simple—easy to deploy, easy to share, and easy to trust.

5.4.6: Final Thought: Don't Drown in Data

With these tips, you can avoid the pitfalls of data chaos. Lightweight Architectural Analytics doesn't have to be expensive or overly complex. It just needs to be:

- Honest
- Relevant
- Usable
- And *maintained with care*

If nothing else, these practices may save you from your next “How did this get so bad?” meeting—or at least provide you with a few laughs along the way.

¹⁶<https://zeljkoobrenovic.github.io/grounded-architecture-dashboard-examples/>

5.5: Tips for Using Lightweight Architectural Analytics

Lightweight Architectural Analytics can generate vast amounts of data. It's akin to having an **atlas of your entire digital landscape**—great for orienting yourself and identifying opportunities. However, maps alone don't provide direction; it's the mindset you adopt while interpreting them that truly matters.



image by cofotoisme from istock

Using analytics is like solving a mystery. The data contains answers—but only if you ask the right questions. With the right mindset, you can transform *information overload* into *architecture superpowers*.

Here are some detective-style questions to help make sense of architectural data:

5.5.1: Are We All Rowing in the Same Direction?

Utilize code overviews, cloud usage explorers, or tech radars to identify **misalignments among teams and systems**. Diverging technology stacks, inconsistent platform usage, or duplicated functionality can lead to **productive debates** and drive real improvements.

5.5.2: Are We Making the Most of Our Technology?

By comparing usage trends across teams, you may uncover **hidden virtuosos**—those who innovate with limited resources—and **bottlenecks**, where adoption is low despite high investment. Both reveal valuable insights.

5.5.3: Do Our Systems Need a Little TLC?

Analytics can highlight:

- Oversized systems
- Rampant duplication
- “God files” that dominate commit histories

These are indicators that part of your architecture may be due for a health check—or a rewrite.

5.5.4: Is More Really More?

For example, if the number of Git merges increases proportionally with team size, while delivery speed does not improve—something is amiss. Scaling can lead to better output, but sometimes it just creates a larger digital mosh pit.

5.5.5: Are We Collaborating the Way We Want To?

Analyzing repository and commit data can reveal **team dynamics** and **coupling patterns**. You’ll discover who is truly collaborating—and who might be unintentionally stepping on each other’s toes.

5.5.6: Are We Working on What We Want to Work On?

We often claim we want to **innovate**, but if the data shows we are bogged down in **legacy maintenance**, it’s a wake-up call. Analytics provides the evidence needed to **challenge strategy** with facts, rather than just opinions.

5.5.7: The Final Question

The data is available, and the story is waiting to be uncovered. So, what's your question?

5.6: To Probe Further

- Online Appendix Software Tools: Examples and Screenshots¹⁷ screenshots of concrete tools I built as a part of Lightweight Architectural Analytics websites.
- Online Appendix Building Lightweight Architectural Analytics¹⁸ a few practical tips on building lean architecture dashboards and documents using simple, widely available tools.
- Open-source architecture dashboard examples¹⁹
- Sokrates²⁰, an open-source polyglot source code examination tool

¹⁷<https://grounded-architecture.io/screenshots>

¹⁸<https://grounded-architecture.io/data-website>

¹⁹<https://zeljkoobrenovic.github.io/grounded-architecture-dashboard-examples/>

²⁰<https://sokrates.dev>

5.7: Questions to Consider

Using data can significantly improve the efficiency and impact of an architecture practice. Ask yourself the following questions:

- *What steps would you take to create an Lightweight Architectural Analytics in your organization?*
- *Are there untapped data sources within your organization that could inform your architectural decisions?*
- *How could you automate gathering data for architectural insights in your organization?*
- *What examples can you provide of the data you've used to gain reliable information about technology in your organization?*
- *How would you examine public cloud billing reports, incident reports, or key business metrics for architectural insights?*
- *How can you ensure your data is reliable and up-to-date?*
- *Do you collaborate with finance and governance teams to incorporate financial and vibrancy data into your data analysis?*
- *Is there a culture of transparency in your organization?*

6: Collaborative Networks



image by mostafa meraji from pixabay

IN THIS SECTION, YOU WILL: Understand that an architecture practice is all about people and get tips on creating organizational structures that support a practical IT architecture practice.

KEY POINTS:

- Developing an architecture practice requires having competent, empowered, and motivated architects. An architecture practice must carefully organize, empower, and leverage scarce talent.
- In my work in the past few years, I combined two teams of architects: a small central architecture team and a cross-organizational distributed virtual team.

Good architects are a rare breed.

They bridge the gap between **business, product, technology, and organizational complexity**. They're the Swiss Army knives of the tech world—part strategist, part engineer, and part diplomat. Hiring architects is akin to searching for a unicorn that can code in Python, translate vision into architecture, and navigate a cross-functional meeting with grace.

Why is this the case? Because effective architects require more than just deep technical expertise. They also need **domain-specific context, organizational awareness**, and the ability to build **trusted relationships** across the company.

So no, you can't simply 3D print architects or hire them in bulk. But you *can* **organize, empower, and amplify the impact of the talent** you already have.

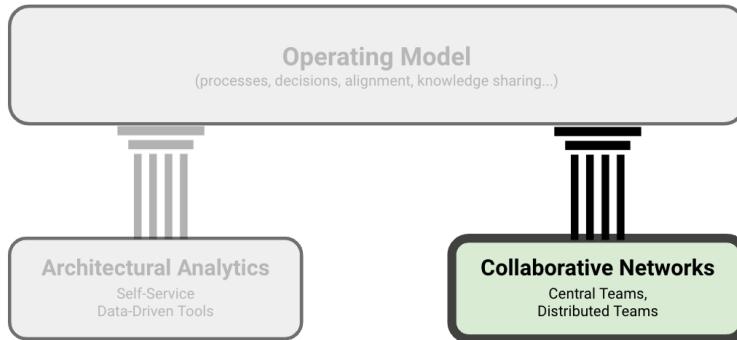


Figure 1: *The Grounded Architecture framework – Collaborative Networks.*

6.0.1: Strong Architecture = Strong Architects

And strong architects don't work alone.

Architecture isn't just a function—it's a **networked capability**. That's why the second pillar of Grounded Architecture focuses on people: **Collaborative Networks**.

In my recent experiences, I worked with two interconnected layers of architectural talent:

6.0.2: Central Architecture Team

Think of this team as the **guides and stewards** of the overall practice. This small group provides:

- Strategic direction
- Methodological support
- Curated data and tools
- Mentorship for others in architecture-related roles

They maintain the big-picture view, uphold architectural quality, and support—not control—the rest of the organization.

6.0.3: Distributed Virtual Architecture Team

This is the **real engine of change**.

These are tech leads, senior engineers, and domain experts embedded within teams across the organization. They're **on the ground**, addressing local problems while remaining connected across various domains. They:

- Raise transparency
- Connect across silos
- Act as change agents
- Scale architectural thinking through influence rather than hierarchy

6.0.4: Why You Need Both

By combining the clarity and support of a central team with the scale and diversity of a distributed team, you cultivate an architectural capability that is:

- **Resilient** (not dependent on one team or person)
- **Scalable** (capable of reaching hundreds of teams)
- **Grounded** in everyday realities
- **Connected** across strategic and technical layers

This is what I mean by a **Collaborative Network**—a structure that empowers architects to be both local and global, strategic and hands-on, leaders and listeners.

It transforms individual expertise into **organizational capability**.

6.1: Background: Centralized vs. Federated Architecture Practice

Most IT architecture practices follow one of two foundational models: **centralized** or **federated**. These models define how architectural responsibilities are distributed, how decisions are made, and how architects collaborate with delivery teams (see Figure 2, adapted from [McKinsey, 2022]).

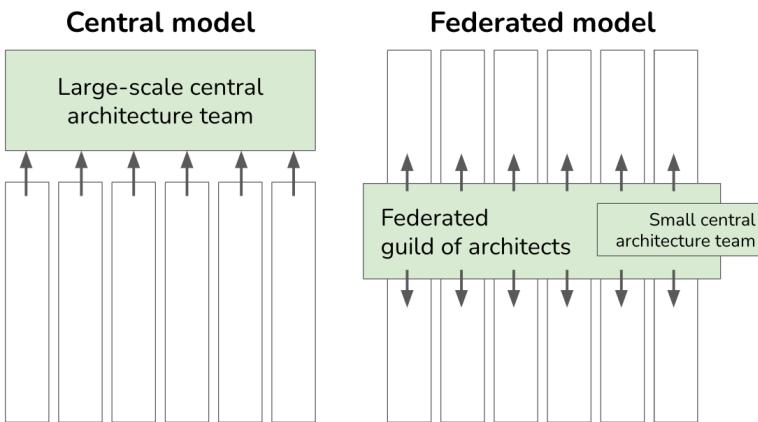


Figure 2: Central vs. Federated Architecture Practice.

6.1.1: The Centralized Model

In a centralized model, a **large central architecture team** governs most architectural decisions. This team sets standards, approves solution designs, and takes primary responsibility for infrastructure, operations, and security.

Development teams depend heavily on this central team for architectural direction, reviews, and implementation guidance. The advantage of this model is **strong governance, consistency, and control**, particularly in regulated or complex environments. However, it can also lead to

slowdowns, bottlenecks, and disconnects between architecture and day-to-day development.

6.1.2: The Federated Model

In the federated model, a **smaller central team**—often referred to as an **Architecture Center of Excellence (CoE)**—provides high-level strategy, shared principles, and support, but the **execution of architecture is distributed**.

Architects are embedded within product, platform, or domain teams. They work closely with delivery teams, supporting planning, technical decisions, and the long-term health of systems from within. This model emphasizes **empowerment, speed, and context-awareness**.

It is especially common in organizations that implement **DevOps practices** and have a **cross-functional team structure**, where architecture is viewed as a shared responsibility integrated tightly into delivery.

6.1.3: Trade-Offs and Trends

Centralized Architecture	Federated Architecture
Strong global governance	High local autonomy and flexibility
Easier to enforce consistency	Faster decision-making in context
Risk of bottlenecks and detachment	Risk of fragmentation and misalignment
Clear oversight and accountability	Stronger alignment with team-level realities

In practice, many organizations adopt a **hybrid model**, which leverages centralized clarity and governance while allowing local teams to operate independently and effectively. This is where **Collaborative Networks** excel: they connect central architecture leadership with distributed practitioners across the organization to ensure both alignment and agility.

6.2: The Hybrid Model

In complex organizations, simply defining architecture responsibilities is not enough; you must **intentionally place the right people in the right roles**. From my experience, the most effective structure combines the strengths of both centralized and federated models:

- A small central architecture team
- A network of architecture guilds and virtual architecture teams

This hybrid model goes beyond a lightweight Center of Excellence (CoE). The central team takes on a proactive, enabling role—not just providing support on demand. It establishes structure, continuity, and alignment, while distributed teams offer reach, scale, and local insight.

6.2.1: A Coordinated Ensemble

You can think of the hybrid model as a **symphony**:

- The **central team** serves as the conductor—coordinating, setting the rhythm, and ensuring harmony.
- The **guilds and virtual teams** are the skilled musicians—each playing their part in local contexts but aligned with the overall composition.

Individually, they can perform successfully. Together, they create a more coherent and scalable architecture practice:

- **Guilds and virtual teams** enhance reach by involving more individuals in architecture discussions. They help scale alignment and surface insights from across the organization.
- The **central team** acts as a **catalyst**—connecting the dots, ensuring strategic coherence, and supporting distributed teams with tools, analytics, and cross-organizational relationships.

6.2.2: Central Architecture Team

Roles within the central team may vary, but it should never be treated as an isolated command center.



Central Architecture Team

image by simonkr from istock

Instead, it should:

- **Build and maintain Lightweight Architectural Analytics:** The system will not operate itself—especially not with emerging AI tools. Ownership, curation, and maintenance are essential.
- **Promote data-informed decision-making:** It is not enough to simply have data; you must advocate for its use. The central team should be the **example-setters** in integrating analytics into real decisions.
- **Connect stakeholders:** Architects need to serve as cross-organizational connectors—building bridges between departments, teams, and leadership layers.

- **Support community-building efforts:** Guilds and distributed teams require coordination, rituals, and support. The central team should drive these initiatives and ensure continuity when participation wanes.

6.2.3: Architecture Guilds & Virtual Architecture Teams

Architecture communities—such as guilds, working groups, or virtual teams—are essential for any federated or hybrid architecture practice.



Architecture Guilds

image by sdi productions from istock

These communities typically include tech leads, staff engineers, and platform owners who:

- Act as architects within their domains
- Collaborate across teams and silos
- Mentor others and share architectural knowledge

- Drive best practices and surface challenges

They function as your **peer-to-peer learning and alignment engine**, helping to scale architectural thinking throughout the organization.

6.2.3.1: Types of Communities

As your guilds expand, consider organizing them into various focus areas:

- **General/Core architecture teams:** Address broad, cross-cutting topics
- **Specialist communities:** Concentrate on specific stacks (e.g., mobile, cloud, frontend)
- **Strategic initiative groups:** Align on larger themes (e.g., cloud migration, platform consolidation, data strategy)

6.2.3.2: Routines for Collaboration

To connect central and distributed architecture efforts, structured collaboration is necessary:

- **Regular forums (e.g., bi-weekly):** Share updates, raise questions, and propose architectural suggestions
- **Summits (annual or bi-annual):** Bring people together to reflect, align, and learn
- **Ad hoc deep-dive workshops:** Address specific problems and explore new patterns collaboratively

The goal is to shift from passive attendance to **active participation**. Architecture should be **co-created**, not handed down.

6.2.4: Architecture Is a Team Sport

Even the best frameworks will fail without a strong network of engaged, empowered individuals behind them. In a hybrid model, everyone has a role to play, and the central team exists to ensure those roles remain aligned and mutually reinforcing.

So, roll up your sleeves. Participate. Connect. Lead. Because great architecture isn't built in silos; it's built **together**.

6.3: Tips for Building Collaborative Networks

Every organization is unique, but several practices have consistently worked well for me when it comes to forming strong architecture teams and building collaborative networks. Whether you are just starting out or evolving an existing practice, here are some practical tips to guide your approach:

6.3.1: Start with the People Already Doing the Work

Before proposing significant organizational changes, **identify and connect with the people already engaged in architecture work**, regardless of their titles. Staff engineers, tech leads, platform owners, and solution experts often perform architectural roles informally.

Bringing these individuals together is never a wasted effort. It lays the groundwork for trust, alignment, and a shared understanding of architectural priorities.

6.3.2: Build a Team, Not Just a Community

If part of your architecture strategy includes creating a **virtual team**, go beyond forming an informal community of practice. Define **clear roles, responsibilities, and expectations**. Establish routines and rituals that foster accountability.

Architecture guilds function best when members know they are not just “showing up,” but actively contributing to something with **impact and ownership**.

6.3.3: Engage Outside the Architecture Circle

Strong collaborative networks extend beyond architects. **Connect early with stakeholders outside of architecture**—including product leaders, engineering managers, operations, data specialists, and finance personnel.

You will need their **support, insights, and buy-in** to create a practice that is integrated rather than isolated. The earlier you involve them, the stronger your network will become.

6.3.4: Grow from Within

Avoid hiring what Gregor Hohpe refers to as a “digital hitman.” These are external experts brought in to “fix” the architecture in isolation.

Instead, **invest in developing internal talent**—individuals who already understand your systems, culture, and context. The best architects combine **technical depth, domain fluency, and organizational awareness**, which cannot be acquired overnight.

6.3.5: Externalize Your Work

Don’t work in isolation. Share what you are doing—both inside and **outside the company**.

- Participate in industry events.
- Publish blog posts or open-source tools.
- Invite feedback on your approach.

Not only does this enhance your practice’s **credibility and influence**, but it also helps you **attract top talent**. When you showcase your architectural work, you become a magnet for others who wish to learn, contribute, and grow. Everyone wants to join the band when you are rocking the stage.



image by chantellev from pixabay

6.4: To Probe Further

- Agile and Architecture: Friend, not Foe¹, by Gregor Hohpe, 2020
- Crafting the optimal model for the IT architecture organization², by Christian Lilley et al., 2022
- Developers mentoring other developers: practices I've seen work well³, by Gergely Orosz, 2022

¹https://architectelevator.com/transformation/agile_architecture/

²<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/tech-forward/crafting-the-optimal-model-for-the-it-architecture-organization>

³<https://blog.pragmaticengineer.com/developers-mentoring-other-developers/>

6.5: Questions to Consider

It is difficult to overestimate the importance of people for an architecture practice, yet many organizations take architectural talent for granted. To reflect on the importance of carefully organizing, empowering, and leveraging scarce architecture talent, ask yourself the following questions:

- *Do you have a strong network of architects across the organization?*
- *Which central, federated, or hybrid model best represents your current an architecture practice? Why was this model chosen, and how effective has it been for your organization?*
- *If you are part of a central architecture team, how would you support the rest of the organization? How would you contribute to the global an architecture practice if you were part of a distributed virtual team?*
- *Consider having the roles of central architecture teams and federated architecture teams in your organization. How would they complement each other?*
- *How effective is the current division of responsibilities among architects in your organization? Are there areas of overlap or gaps in coverage?*
- *What steps has your organization taken to ensure architects are well-connected across all parts and levels? What impact has this had on transparency and the implementation of changes?*
- *Reflect on the diversity of team structures within your organization. How does this diversity impact the roles and responsibilities of architects?*

7: Operating Model: General Principles

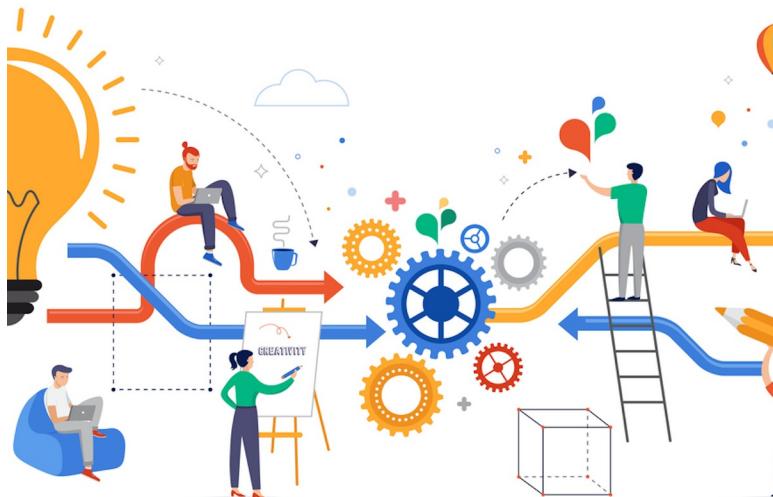


image by ma_rish from istock

IN THIS SECTION, YOU WILL: Understand what activities you can do as a part of an architecture practice and get tips on creating pragmatic operating models for an architecture practice.

KEY POINTS:

- The Operating Model is a system of processes and agreements enabling architects to do everything architecture typically does, leveraging Lightweight Architectural Analytics and Collaborative Networks to create a data-informed, organization-wide impact.
- Examples of activities include: supporting teams in their daily work; tracking tech debt; performing technical due diligence; standardizing processes and documentation; defining cloud, data, and platform strategies.

Each organization will have different architectural needs and contexts. When forming an architecture practice, I use as a starting point these [two pieces of advice from Gregor Hohpe](#)¹:

- *“Your architecture team’s job is to **solve your biggest problems**. The best setup is the one that allows it to accomplish that.”*
- *“Your organization has to earn its way to an effective an architecture practice. You can’t just plug some architects into the current mess and expect it to solve all your problems.”*

Considering Gregor Hohpe’s previous two points, I approach defining an architecture practice with the mindset that there is no one-size-fits-all method. You must find your own activities and operating models to enable architecture to solve the most critical problems.

No matter which operating models you select, it’s crucial to develop **explicit agreements** and “rules of engagement” with key stakeholders. This collaborative approach is essential to create a sustainable and practical an architecture practice.

¹<https://architectelevator.com/architecture/organizing-architecture/>

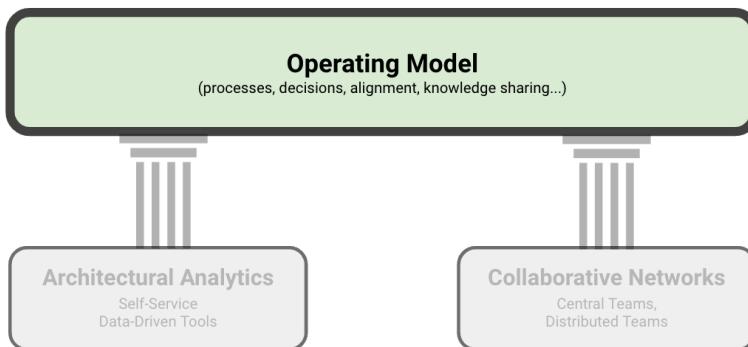


Figure 1: The Grounded Architecture framework: Operating Model.

This section outlines some lessons I learned when defining IT architecture operating models. The Operating Model (Figure 1) is a part of the Grounded Architecture framework that defines a set of **processes and agreements** that allow architects to do everything an architecture practice typically does. The model should leverage Lightweight Architectural Analytics and Collaborative Networks to develop a data-informed, organization-wide impact. Lightweight Architectural Analytics and Collaborative Networks provide a basis for data-informed decision-making that is well-embedded in the organization.

7.1: Examples of Architecture Activities

An Operating Model enables a structured and strategic approach to an architecture practice within the organization.



image by brauns from istock

Here are examples of the activities I have been engaged in with architects to provide a clearer understanding of what I mean by an operating model.

- **Designing Mechanisms for Teams to Make Better Decisions:** These mechanisms involved creating global decision-support frameworks such as advisory forums facilitating informed discussions across teams. For compliance-sensitive projects, we establish formal design authorities. Additionally, we develop team-specific mechanisms, like escalation paths, to resolve decision conflicts effectively (e.g., when teams disagree on a common messaging middleware).
- **Supporting Teams in Their Daily Work:** This support entailed integrating into key team activities and aligning architectural work with team rituals to provide timely support. We assisted teams during all critical phases, such as reviewing architecture proposals

before the commencement of a project or sprint, ensuring alignment with overall architectural standards.

- **Supporting Planned New Initiatives and Projects:** Ensuring seamless alignment between projects that require multi-team collaboration is crucial. We worked to facilitate communication and coordination, ensuring all teams are on the same page regarding project goals and requirements.
- **Supporting Teams in Dealing with the Legacy Landscape:** We provided data and insights about the legacy landscape, identifying problematic areas such as frequently changed, low-quality, untested legacy code. We helped define scenarios and roadmaps for legacy modernization, ensuring a structured approach to updating and maintaining legacy systems.
- **Tracking Tech Debt and Defining Tech Debt Reduction Programs:** This involves creating a centrally aligned backlog of technical debt and defining programs for its reduction. We integrate these programs into the planning processes to ensure that tech debt is managed proactively and effectively.
- **Performing SWOT and Other Analyses of Platforms and Systems:** Conducting deep dives to understand specific areas of the technology landscape. We performed SWOT (Strengths, Weaknesses, Opportunities, Threats) analyses and other assessments. These analyses helped in creating comprehensive plans and roadmaps for improvement.
- **Standardizing Processes and Documentation:** We defined standard templates for key documents such as Architectural Decision Records (ADRs), Technical Design Reviews (TDRs), and common diagrams. This standardization ensures consistency and clarity across all architectural documentation.
- **Supporting Merger and Acquisition (M&A) Activities with Expertise and Analyses:** We provided analyses, recommendations, and integration planning for mergers and acquisitions. Such support ensures that architectural considerations are well-integrated into M&A activities, facilitating smoother transitions and integrations.
- **Defining Key Technology Strategies:** We contributed to the development of essential technology strategies, including those for Cloud, Data, and Platforms. These strategies provide a clear

roadmap for technological development and investment, ensuring alignment with business goals.

- **Defining Vision and Direction of Technology:** In collaboration with Engineering Leaders, we created a sustainable organizational setting that aligns with the overarching technology strategies. This work involved setting a clear vision and direction for the technology landscape within the organization.

7.2: Guiding Principles for Architectural Excellence: Policies, Autonomy, and Engagement

In this section, I address different guiding principles of architectural work:

- Our **operating framework** always emphasizes a **collaborative** and **supportive** approach. Architects should empower development teams to make most decisions while ensuring strategic alignment and minimal compatibility. Architects should engage early in processes to **avoid bureaucratic delays**, focus on constant motion between daily support and strategic tasks, and use data to inform decisions.
- The **distributed decision-making** model promotes team autonomy complemented by high transparency and alignment, guided by principles that balance autonomy with global consistency.
- The “Golden Paths” concept enhances uniformity and efficiency.

7.2.1: High-Level Operating Framework

While exact activities and their scope will depend on an organization setting and will change over time, I usually followed a common operational framework in daily work inspired by Gregor Hohpe’s strategy-principles-decisions model (Figure 2).

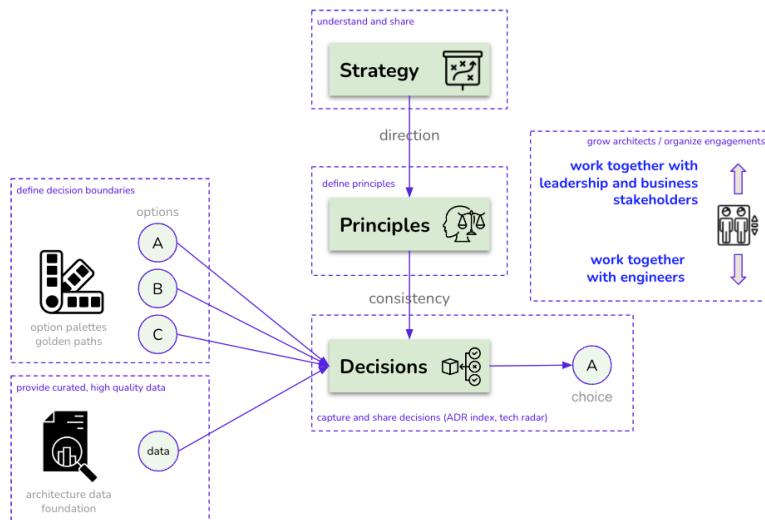


Figure 2: A common operating framework I typically use for Grounded Architecture activities.

Here are the key characteristics of this operating framework:

Engagement mindset:

- Architects engage with stakeholders and teams in a **collaborative and supportive manner**.
- Architects aim to **empower the teams** so that they make most of the decisions.

Contributions of architects:

- Bring relevant data to inform decisions leveraging [Lightweight Architectural Analytics](#).
- Define decision boundaries to enable minimal compatibility and strategic alignment (e.g., golden paths or tech stack constraints).
- Define fundamental principles to facilitate consistency in decision-making.
- Share and generalize lessons learned via [Collaborative Networks](#).

Social dynamics of architects:

- Architects spend their time in **constant motion** between supporting teams' **daily work** and working on **strategic topics**, helping the organization achieve alignment between strategy and implementation.

Shift left:

- Avoid **formal bureaucratic approval processes**, where architects appear too late and are frequently busy approving trivial decisions.
- Have architects **involved early** in any of the processes, such as during the planning and preparation stages, where it is possible to make more significant changes. Think of it as having the architects as early birds catching the architectural worms, making big changes before the day officially starts.

7.2.2: Distributing Decisions, Autonomy, and Alignment

With any operating model, I aim to keep architectural decision-making distributed across the organization and embedded in the development teams. Development teams traditionally have the best insights and most information relevant for making decisions. As noted by Gregor Hohpe, the worst case of organizational decision-making happens when people with relevant information are not allowed to make decisions, while people who lack sufficient information make all decisions. Grounded Architecture aims to make relevant information more readily available to a broader audience and better connect people when making decisions.

While I aim to create a mechanism to give teams autonomy, autonomy does not mean that teams are alone, do not align with anyone, do not get feedback from anyone, and do whatever they want. Teams must complement **autonomy** with high **transparency** and **proactivity** in alignment with other groups.

I have sometimes implemented the concept of a **decision pyramid** (Figure 3) to give the teams **maximal autonomy** while maintaining a **minimal** level of **global alignment** and compatibility.

The **decision pyramid** highlights that development teams should make most decisions. However, several strategic and area-level choices may

provide team decision boundaries. For example, selecting the public cloud provider is typically a CTO-level strategic decision. Similarly, engineering leaders in some areas may want to limit some choices, such as the number of programming languages, to more easily train new people, maintain code, and support moves between teams.



Figure 3: A decision pyramid. The development teams should make most decisions. However, several strategic and area-level decisions may provide decision boundaries for teams (e.g., golden paths or tech stack constraints).

7.2.3: General Architecture Decision Policy

Distributed decision-making scales well, but it can lead to chaos if entirely uncoordinated. Some decision policies are needed. Inspired by the famous [Netflix expense policy](#)², “Act in Netflix’s best interests”, I frequently argued that architecture decision policy could similarly be summarized in six words: “Decide in the Organization’s Best Interests.”

²<https://hbr.org/2014/01/how-netflix-reinvented-hr>

“Decide in the Organization’s Best Interests.”

What I mean by that is that **anyone can make architecture decisions**, provided that, in addition to their specific requirements, they also think about the **impact of their choices** on:

- **Overall organizational complexity:** Technology is more manageable by limiting tech diversity, size, and dependencies. Limiting technology choices reduces the attack surface with fewer third-party dependencies and tool ecosystems (build, testing, etc.).
- **Ease of moving people** between teams (both to get help and help others): Do not unnecessarily create exotic islands with few experts in technologies not supported or widely used in the organization. People cannot get help or move across the organization as their expertise may be useless outside the team.
- **Ease of training and onboarding** of internal and external developers: Using conventional technologies supported by external learning resources (e.g., books, tutorials) significantly helps find and grow experts.
- **Talent density** and the possibility of performing at the world-scale level: Building world-scale technology and scaling requires in-depth knowledge and fine-tuning. You cannot achieve it with only a few in-house experts.
- **New reorganizations:** If the ownership of components changes (e.g., another team is taking it over), would your choices fit with other components from other areas?
- Reducing global **duplication of effort** and inefficiencies: Are you doing the work others are doing? Can others reuse your work? Can you reuse the work of others?

While it may not always be enough, this simple policy can resonate well

with many people and can encourage them to be more thoughtful when making decisions.

7.2.4: Golden Paths

I have found that the concept of Golden Paths provides an excellent ground to **drive alignment and collaboration** in architecture activities. Golden Paths is an approach utilized to streamline and unify the development process within a software ecosystem, aiming to tackle fragmentation and foster consistency, inspired by [Spotify's implementation³](#). Golden Paths can be described as “**opinionated and supported**” routes developers can follow to build systems efficiently and effectively.

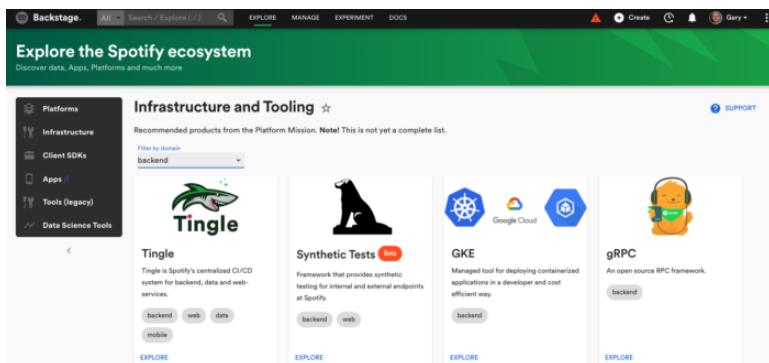


image by engineering.atspotify.com

Golden paths provide a solid **foundation for aligning** architecture activities, serving as a common target of work for Guilds and central architectural teams. Rather than being solely knowledge-sharing entities, **guilds** can be empowered to **develop golden paths**, serving as an excellent catalyst for more effective community engagement. This approach not only enhances the role of guilds but also increases the adoption of golden paths as they are created collaboratively.

Golden Paths can be crucial to an organization’s IT development landscape as a deliberate and strategic effort to promote **uniformity, effi-**

³<https://engineering.atspotify.com/2020/08/how-we-use-golden-paths-to-solve-fragmentation-in-our-software-ecosystem/>

ciency, and reliability. By advocating for a set of preferred technologies and practices that are **well-supported, secure**, and aligned with the organization's broader objectives, Golden Paths can guide developers to build less fragmented, and faster-to-develop software. Ultimately, this leads to higher-quality and more maintainable IT systems.

7.3: Embracing Diversity

When building architecture guilds and virtual architecture teams, it's crucial to acknowledge that organizational units have diverse structures and sizes. In big organizations, **embracing diversity** is a prerequisite to having a broad impact.

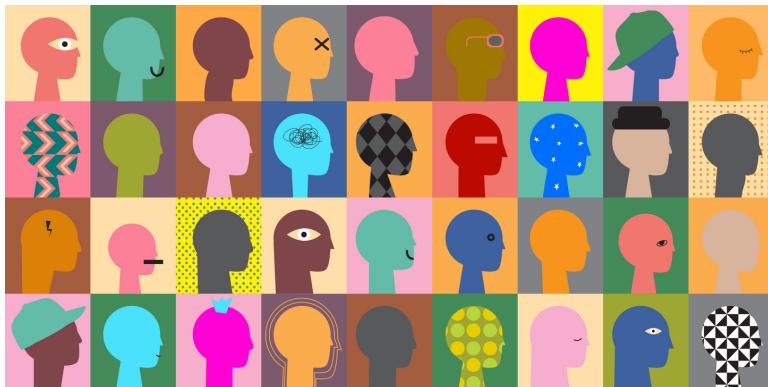


image by annaspoka from istock

There is no one-size-fits-all solution for assigning architecture responsibilities within departments. Based on Gregor Hohpe's view of architects and their teams' relationships, I've generally encountered three types of team-architect systems:

1. **Benevolent “dictator”:** An architect or architecture team tells developers what to do or how to do things. The key nuance here is whether the communication is unidirectional or bi-directional.
2. **Primus inter pares (first among equals):** Architects are embedded within teams where each is just another team member, but with a focus on the system structure and trade-offs and taking a longer-term view than other team members.
3. **Architecture without architects:** Architecture is done within teams, but the task is a shared responsibility among multiple (or all) team members. This approach is often the preferred model in modern technology organizations.

Remember, there is no magic bullet. Different structures work for various organizations; sometimes, the best solution is a mix of these approaches.

7.4: Setting Boundaries

One of the amusing challenges with setting up an architecture practice in an organization is that everyone seems to have a different idea of what “architecture” should entail. It’s like asking people to describe a unicorn: some imagine a mythical, majestic creature, while others picture a sparkly horse with a horn that grants wishes. Good architects can do many things, but this versatility might not always be the most effective way to support the organization. We need to **set boundaries** so that we can focus on what’s important rather than becoming frazzled by what’s not.



image by ingenui from istock

To be effective, I’ve found it crucial to establish and clearly communicate some “**rules of engagement**” (ROE). Think of ROE as the office playbook for how architects should operate. In a corporate setting, ROE are the principles that guide how employees and departments interact with each other, clients, and stakeholders. This includes communication protocols, decision-making processes, and conflict-resolution mechanisms. Essentially, ROE sets the stage for what’s expected and what’s not, ensuring everyone plays nicely and fairly.

While you may need to tailor these rules to fit your organization, I found it helpful to set expectations for what the team should be able to do to qualify for the architecture support. Here's a handy list of expectations for teams seeking architecture support. This also helps clarify what an architecture practice isn't supposed to do:

1. **Organizational Awareness and Connections:** Teams should know all relevant stakeholders and actively engage with them. This knowledge should include product, development, and business stakeholders. Planning should be collaborative across all affected teams, with active working relationships with global functions like QA, DevOps, or Security.
2. **Enough Capacity and Skills:** Teams should have adequate development capacity with the right skills and seniority to innovate and maintain their products.
3. **Strategic Awareness:** Teams should understand the organization's strategic goals, technologies, and other relevant strategies, and know their role within these frameworks.
4. **Technical Documentation Literacy:** Teams should be capable of creating technical documentation, such as ADRs (Architecture Decision Records) or RFCs (Request for Comments).
5. **Technology Standard Awareness:** Teams should be familiar with the organization's technology standards, including golden paths and guidelines for planning, documentation, security, DevOps, and QA processes.
6. **Participation and Citizenship:** Teams should actively participate in relevant communities (like architecture guilds) and global events (such as architecture summits).
7. **Tech Debt Management:** Teams must be aware of the technical debt they create and maintain, ideally having a tech debt backlog and a plan for "paying" it back.

Aligning on these rules with the teams helps ensure productive conversations about architectural support. When these conditions are met, an architecture practice can help teams level up. When they're not, architecture support can't be as effective. However, that doesn't mean struggling teams are left in the lurch. Architecture can help teams meet these expectations but can't compensate for their total lack. Teams need to take the initiative and lead. For instance, it's impractical to have architects

working full-time for months with one team as their senior developer. However, architects can coach and help developers grow which is a more scalable approach. Similarly, architects can assist in building relationships with other teams, but the teams themselves need to be active and engaged.

So, set those expectations, establish your rules of engagement, and watch as your architecture practice goes from a sparkly unicorn to a well-oiled machine!

7.5: To Probe Further

- Scaling the Practice of Architecture, Conversationally⁴, by Andrew Harmel-Law, 2021
- Scaling Engineering Teams via RFCs: Writing Things Down⁵, by Gergely Orosz, 2022
- Transformation Agents: An Engagement Model⁶, by Gregor Hohpe, 2022
- Would you like architects with your architecture?⁷, by Gregor Hohpe, 2021

⁴<https://martinfowler.com/articles/scaling-architecture-conversationally.html>

⁵<https://blog.pragmaticengineer.com/scaling-engineering-teams-via-writing-things-down-rfc/>

⁶<https://architectelevator.com/transformation/transformation-engagement-model/>

⁷<https://architectelevator.com/architecture/organizing-architecture/>

7.6: Questions to Consider

Your an architecture practice job is to solve the biggest problems in your organization. Ask yourself the following questions:

- *How can you identify the most critical problems that your architects need to solve in your organization?*
- *What activities and operating models can you think of that will best enable architecture in your organization to work on these critical problems?*
- *What does the Operating Model look like in your organization, and how could it be improved?*
- *Which of the provided examples of architectural activities are you currently performing in your organization?*
- *How does the proposed common operating model align with your current operational practices in your organization? What changes might be necessary to adopt this model?*
- *In your experience, how early are architects involved in projects and activities? Do you agree with the goal of 'shifting left' the architecture work?*
- *How are architectural decisions distributed across your organization currently? How could this process be improved to ensure the people with the most relevant information make the decisions?*
- *How could you better implement a mechanism to give teams autonomy while maintaining alignment and compatibility with global strategy?*
- *How does the concept of a decision pyramid resonate with you?*
- *Which strategic and area-level decisions provide team decision boundaries in your organization? Are there areas where you need more or less limitations to optimize performance?*

8: Cooperation-Based Operating Model: Six Simple Rules



image by nanostockk from istock

IN THIS SECTION, YOU WILL: Get an introduction to Six Simple Rules, a model for setting up organizational structures based on cooperation.

KEY POINTS:

- The Six Simple Rules approach emphasizes that in today's complicated business environment, you must set up organizational structures based on cooperation.
- To deal with complexity, organizations should depend on the judgment of their people and on these people cooperating.
- This view is well aligned with the ideas of Grounded Architecture.

The book [Six Simple Rules: How to Manage Complexity without Getting Complicated](#)¹ by Yves Morieux and Peter Tollman offered fresh air for my vision of an architecture practice. Morieux and Tollman introduced the concept of **Smart Simplicity** with six rules or strategies that enable organizations to promote new behaviors and improve performance. The Six Simple Rules approach emphasizes that in today's business environment, you need to set up **organizational structures based on cooperation**.

The authors developed the Six Simple Rules approach as a practical solution for today's complex business environment. The rules in the book are based on the premise that the key to managing complexity is the **combination of autonomy and cooperation**. The book advocates for the setup of organizational structures that harmonize, empowering individuals with more autonomy to act. This approach is about trusting the capabilities of the organization's people to handle complex problems, fostering a cooperative and efficient work environment.

In this chapter, I explore how Grounded Architecture and Six Simple Rules are best friends. The Six Simple Rules ideas have been a significant source of inspiration for my work. [Conway's Law](#)² shows that the link between organizational structures and IT architecture is like peanut butter and jelly—strong and better together. The Six Simple Rules approach and architectural work are all about managing complexity without getting tangled up.

¹<https://www.bcg.com/capabilities/organization-strategy/smart-simplicity>

²<https://martinfowler.com/bliki/ConwaysLaw.html>

8.1: Background: Hard and Soft Management

One of the Six Simple Rules' central premises is that **conventional management approaches**, which the authors split into hard and soft, are neither sufficient nor appropriate for the complexity of organizations nowadays.

8.1.1: Hard Approach

The **hard approach** rests on two fundamental assumptions:

- The first is the belief that **structures, processes, and systems** have a direct and predictable effect on performance, and as long as managers pick the right ones, they will get the performance they want.
- The second assumption is that the **human factor is the weakest and least reliable link** of the organization and that it is essential to **control people's behavior through the proliferation of rules** to specify their actions and through financial incentives linked to carefully designed metrics and key performance indicators (KPIs) to motivate them to perform in the way the organization wants them to.

When the company needs to meet new performance requirements, the **hard response** is to add new **structures, processes, and systems** to help satisfy those requirements. Hence, introducing the innovation czar, the risk management team, the compliance unit, the customer-centricity leader, and the cohort of coordinators and interfaces have become so common in companies.



image by cyano66 from istock

In my experience, IT architecture support in organizations following a hard management approach has the following characteristics:

- **Heavy Reliance on Tools:** The architecture would focus on tools, automation, and workflows that **enforce compliance**, standardization, and control. There might be a strong reliance on enterprise resource planning (ERP) systems, centralized data warehouses, and other structured tools that enable **strict adherence** to predefined processes.
- **Security and Compliance:** There would be a significant emphasis on **compliance frameworks**, security protocols, and risk management systems. These systems would be designed to ensure that all actions are traceable, compliant with regulations, and aligned with the company's **predefined rules and processes**.
- **Bureaucratic Systems:** The architecture might include multiple **layers of control** and coordination units, such as a risk management system, compliance databases, and innovation management software. Each of these units would be designed to enforce specific aspects of the organization's performance objectives.

8.1.2: Soft Approach

On the other end, we have a soft management approach. According to the soft approach, an organization is a set of **interpersonal relationships** and the **sentiments** that govern them.

- **Good performance is the by-product of good interpersonal relationships.** Personal traits, psychological needs, and mindsets predetermine people's actions.
- To change behavior at work, you need to **change the mindset (or change the people).**



image by alessandro biascioli from istock

In my experience, IT architecture support in organizations following a soft management approach has the following characteristics:

- **Employee Empowerment:** Systems would likely be designed to empower employees by providing the tools they need to **succeed independently**, such as access to real-time data, **self-service analytics**, and systems that encourage innovation and creativity.
- **Employee-Centric:** The architecture would focus on tools facilitating communication and **knowledge sharing**. These tools might include **collaboration platforms**, social intranets, and user-friendly interfaces that enable employees to interact more effectively.
- **Personalization:** Systems might be more personalized, offering **customization options** that align with individual preferences and

needs. This personalization could include customizable dashboards and personalized workspaces.

8.2: Collaboration Approach

Hard and soft management approaches are limited in today's world and are harmful to cooperation. A **hard approach** introduces **complicated mechanisms**, compliance, and "checking the box" behaviors instead of the engagement and initiative to make things work. The **soft approach's** emphasis on **good interpersonal feelings** creates **cooperation obstacles** as people want to maintain good feelings.

The Six Simple Rules approach emphasizes that in today's business environment, you must set up **organizational structures based on cooperation**. More specifically, the Six Simple Rules approach involves the interplay of **autonomy** and **cooperation**. The authors emphasize the critical difference between autonomy and self-sufficiency. **Autonomy** is about fully mobilizing our intelligence and energy to **influence outcomes**, including those we do not entirely control. Self-sufficiency is about **limiting our efforts** only to those outcomes that we **control entirely without depending** on others. Autonomy is essential for coping with complexity; self-sufficiency is an obstacle because it **hinders the cooperation** needed to make autonomy effective.

The first three rules create the conditions for **individual autonomy and empowerment** to improve performance.

- **Understand what your people do.** Trace performance back to behaviors and how they influence overall results. Understand the context of goals, resources, and constraints. Determine how an organization's elements shape goals, resources, and constraints.
- **Reinforce integrators.** Identify integrators—those individuals or units whose influence makes a difference in the work of others—by looking for points of tension where people are doing the hard work of cooperating. Integrators bring others together and drive processes.
- **Increase the total quantity of power.** When creating new roles in the organization, empower them to make decisions without taking power away from others.

This difference between **Autonomy** and **Self-Sufficiency** leads us to the second set of rules that compels people to confront complexity and use their newfound autonomy to cooperate with others so that **overall performance, not just individual performance**, is radically improved.

- **Increase reciprocity.** Set clear objectives that stimulate mutual interest to cooperate. Make each person's success dependent on the success of others. Eliminate monopolies, reduce resources, and create new networks of interaction.
- **Extend the shadow of the future.** Have people experience the consequences that result from their behavior and decisions. Tighten feedback loops. Shorten the duration of projects. Enable people to see how their success is aided by contributing to the success of others.
- **Reward those who cooperate.** Increase the payoff for all when they cooperate in a beneficial way. Establish penalties for those who fail to cooperate.

8.3: Rule 1: Understand What Your People Do

The first rule states that you must genuinely understand performance: **what people do and why they do it**. When you know why people do what they do and how it drives performance, you can define the minimum sufficient set of interventions with surgical accuracy.



image by scyther5 from istock

8.3.1: Guidelines for Understanding Performance

To genuinely understand performance, consider the following principles:

- Trace performance back to behaviors, understanding how these behaviors influence and combine to produce overall results.
- Utilize observation, mapping, measurement, and discussion to gain insights.
- Comprehend the context of goals, resources, and constraints within which current behaviors are rational strategies for people.
- Discover how your organization's elements—structure, scorecards, systems, and incentives—shape these goals, resources, and constraints.

8.3.2: Leveraging Architecture Practice

Architecture practice can significantly aid in understanding organizational behaviors through:

- Establishing a **Lightweight Architectural Analytics** with an overview of various data sources to reveal where activities occur, visible trends, and cooperation among people.
- Utilizing the **Collaborative Networks** to connect individuals and enable them to learn about activities in different parts of the organization.

8.4: Rule 2: Reinforce Integrators

The Six Simple Rules approach emphasizes the importance of reinforcing integrators by looking at those **directly involved in the work**, giving them **power and interest to foster cooperation in dealing with complexity** instead of resorting to the paraphernalia of overarching hierarchies, overlays, dedicated interfaces, balanced scorecards, or coordination procedures.



image by robert owen_wahl from pixabay

8.4.1: Guidelines for Reinforcing Integrators

To strengthen integrators within your organization, consider these strategies:

- **Use emotions** to identify candidates. Feelings provide crucial clues for analysis and can act as symptoms of integration issues.
- **Identify operational units** that can be integrators among peer units due to their specific interests or power.

- Remove managerial layers that do not add value and reinforce others as integrators by eliminating specific rules and relying on observation and judgment over metrics when cooperation is involved.

8.4.2: Enhancing Integrators Through Architecture Practice

Architecture practice can play a vital role in reinforcing integrators by:

- Utilizing the **Collaborative Networks** to help identify and connect integrators, leveraging their work effectively.
- Emphasizing architects as critical integrators and integrator role models within the organization, defining them as essential components of the organizational “*superglue*.”
- Establishing a **Lightweight Architectural Analytics** to support integrators with data and insights, enabling them to perform more informed and effective work.

8.5: Rule 3: Increase the Total Quantity of Power

Whenever you consider an **addition to** your organization's **structure, processes, and systems**, think about **increasing the quantity of power**. Doing so may **save you from increasing complicatedness** and enable you to achieve a more significant impact with less cost. You can increase the quantity of power by allowing some functions to influence performance and stakes that matter to others.



image by prostock_studio from istock

8.5.1: Guidelines for Increasing Power Quantity

To enhance the quantity of power within your organization, consider these actions recommended by the Six Simple Rules approach:

- When making design decisions that could shift the balance between central and unit power, functions, and line managers, ensure that some parts of the **organization benefit from new power bases**.

This approach helps meet complexity requirements and avoids future disruptions from pendulum swings.

- When creating new functions, ensure they have the power to fulfill their roles **without diminishing the power others need** to fulfill theirs.
- When introducing new tools for managers, such as planning or evaluation systems, evaluate whether these tools **act as resources or constraints**. Implementing a few tools creates a critical mass of power, which is more effective than introducing many tools sequentially.
- Regularly **enrich power bases** to maintain agility, flexibility, and adaptability.

8.5.2: Enhancing Power Quantity Through Architecture Practice

Architecture practice can support the increase in power quantity through an operating model that promotes distributed decision-making:

- Through the **Collaborative Networks**, you can enhance decision-making power by distributing architectural decision-making across the organization and embedding it within development teams, which typically have the best insights and most relevant information.
- Additionally, **Lightweight Architectural Analytics**, accessible to all interested members of the organization, can provide data and insights that empower individuals in their daily work.

8.6: Rule 4: Increase Reciprocity

In the face of business complexity, work is becoming more interdependent. To meet multiple and often contradictory performance requirements, **people must rely more on each other**. They need to **cooperate directly** instead of depending on dedicated interfaces, coordination structures, or procedures that only add to complicatedness.



image by natnan srisuwann from istock

8.6.1: Guidelines for Enhancing Reciprocity

Reciprocity involves recognizing that people or units in an organization have a mutual interest in cooperation and that the success of one depends on the success of others. To foster this reciprocity, follow these guidelines:

- **Eliminate monopolies** to ensure no single entity has exclusive control.
- **Reduce resources** to encourage more efficient and collaborative use.

- **Create new networks** of interaction to facilitate better communication and cooperation.

8.6.2: Enhancing Reciprocity Through Architecture Practice

Architecture practice can significantly increase reciprocity within an organization:

- The **Collaborative Networks** supports creating new networks of interactions, directly reinforcing reciprocity.
- A **hybrid operating model** relies on the mutual success of an architecture practice and development teams. Architects' **impact** is essential, and their support depends on the feedback from the groups they assist. Integrating this feedback into architects' performance evaluations is crucial for enhancing reciprocity between architecture and other units.

8.7: Rule 5: Extend the Shadow of the Future

The Six Simple Rules approach emphasizes the importance of making visible and clear **what happens tomorrow as a consequence of what they do today**.



image by joe from pixabay

8.7.1: Guidelines for Extending the Shadow of the Future

The Six Simple Rules approach recommends four strategies to extend the shadow of the future:

- **Tighten the feedback loop** by increasing the frequency of moments when people experience the consequences of the fit between their contributions.

- Bring the endpoint forward by shortening the duration of projects.
- Tie futures together so that successful moves are conditioned on contributing to the success of others.
- Ensure people walk in the shoes they make for others.

8.7.2: Extending the Shadow of the Future Through Architecture Practice

Architecture practice can play a crucial role in extending the shadow of the future through various methods:

- **Lightweight Architectural Analytics** can create transparency and provide the data necessary to model the future. This data can be used to develop simulations and roadmap options.
- Applying **economic modeling** to architecture decision-making helps describe the future consequences of today's actions, directly supporting long-term planning and decision-making.

8.8: Rule 6: Reward Those Who Cooperate

Lastly, the Six Simple Rules approach recommends that when you cannot create direct feedback loops embedded in people's tasks, you need **management's intervention to close the loop**. Managers must then use the familiar performance evaluation tool but in a very different way to reward those who cooperate.



image by stocksnap from pixabay

8.8.1: Guidelines for Rewarding Cooperation

To effectively reward those who cooperate, managers should:

- Go beyond technical criteria and avoid placing blame solely on where the root cause originated. Accept that execution problems arise from various reasons. The smart approach is to reduce rewards for those who **fail to cooperate** in solving a problem, even if the problem isn't in their direct area, and increase rewards for units that cooperate beneficially.

- Avoid blaming failure and instead focus on blaming the inability to help or seek help.
- Use simple questions to shift managerial conversations, making transparency and ambitious targets resources rather than constraints. This approach helps managers act as integrators, leveraging cooperation and rich information to achieve superior results.

8.8.2: Rewarding Cooperation Through Architecture Practice

Architecture practice can facilitate rewarding cooperation by making it easier for individuals to help others and ask for help:

- A strong **Collaborative Networks** can provide the context and networks necessary for easier collaboration.
- Adding diverse data sources to **Lightweight Architectural Analytics** can create transparency about cooperation opportunities and challenges, supporting a more collaborative environment.

8.9: To Probe Further

- Six Simple Rules: How to Manage Complexity without Getting Complicated³

³<https://www.bcg.com/capabilities/organization-strategy/smart-simplicity>

8.10: Questions to Consider

- *How can the concept of Smart Simplicity apply to your current role or position within your organization?*
- *Do you feel the structures, processes, and systems directly and predictably affect performance in your organization?*
- *Do you feel that your organization views the human factor is viewed as the weakest link? How does this affect how you and your colleagues perform?*
- *How do you perceive the balance between your organization's hard and soft management approaches? Is one approach more dominant?*
- *How does your organization currently promote autonomy and cooperation among employees? Are there areas for improvement?*
- *How do the assumptions of hard and soft management approaches hinder cooperation in your organization?*
- *How can you increase the total power within your organization without taking power away from others?*
- *How can your organization increase reciprocity and make each person's success dependent on the success of others?*
- *How can your organization extend the shadow of the future? Are there feedback mechanisms in place to make people accountable for their decisions?*
- *How are those who cooperate rewarded in your organization? Are there mechanisms in place to increase the payoff for all when they cooperate beneficially?*
- *How can architecture practice in your organization support the implementation of the Six Simple Rules?*
- *How do your organization's current systems and structures promote or hinder the cooperation needed to make autonomy effective?*

9: Governance: Nudge, Taxation, Mandates



image by nonbirinonko from pixabay

IN THIS SECTION, YOU WILL: Understand that a technology governance model should be a well-balanced hybrid of three different styles of governing: mandates and bans, taxes, and nudging.

KEY POINTS:

- Architecture practice should support governance models adaptable to organizations' complex and diverse needs. A technology governance model should be a well-balanced hybrid of three different styles of governing: mandates and bans, taxes, and nudging.
- Nudging is a form of governing where you create subtle or indirect suggestions influencing someone's behavior or decision-making without forcing them or limiting their freedom of choice.
- Governing with taxes (economic incentives) is a form of guiding in which people are not forbidden to make some decisions but need to "pay" some form of taxes on used resources.
- With mandates and bans, you guide people by explicitly defining what they should or should not do.

Governance refers to the rules, practices, and processes that dictate how an organization is **directed, managed, and held accountable**. It clarifies who makes decisions, how they are made, and how the outcomes are monitored and enforced.

Governance extends beyond high-level board meetings; it is relevant across various areas of an organization, including:

- **Corporate governance** – guiding leadership, accountability, and strategic direction.
- **IT governance** – ensuring technology investments align with business goals.
- **Project governance** – managing the execution of projects to align with strategic objectives.
- **Data governance** – ensuring data is accurate, secure, and responsibly used.

Every organization that utilizes technology has an IT architecture, which is the blueprint for how its systems, software, and infrastructure work

together. However, IT architecture is more than technical diagrams; it is a form of governance.

IT architecture helps an organization to:

- Standardize systems and processes, thereby reducing complexity.
- Align technology with business priorities and strategies.
- Manage risks related to security, compliance, and organizational change.
- Utilize resources efficiently, avoiding waste and redundancy.
- Support better decision-making by providing clear structures and insights.
- Encourage innovation without sacrificing control.
- Measure performance and facilitate continuous improvement.

In essence, IT architecture translates governance principles into practical, everyday practices that bolster business success through technology.

Governance is not without its challenges. It often has to address:

- Conflicting priorities among departments or teams.
- Rapid changes in technology, regulations, or customer needs.
- Complex environments characterized by multiple systems, vendors, and stakeholders.

For governance to be effective, it must be:

- Adaptable – able to respond to evolving needs.
- Collaborative – engaging diverse voices and perspectives.
- Strategic – balancing immediate demands with long-term visions.

The best governance frameworks are living systems; they evolve as the organization and its environment change.

Architecture, particularly in IT, should facilitate governance models that adapt to the complexity and diversity of the modern organization. One-size-fits-all approaches are rarely successful. Therefore, I advocate for a hybrid governance model—a thoughtful blend of different approaches tailored to specific situations.

This model incorporates three styles of influence:

1. **Nudging** – offering gentle guidance that influences behavior without restricting choice.
2. **Taxes** (or economic incentives) – rewarding positive behavior or discouraging harmful practices through resource allocation.
3. **Mandates and bans** – establishing clear rules and boundaries as necessary.

Each of these approaches plays a significant role. Together, they form a flexible and effective framework for managing how technology is employed and evolved within the organization.

9.1: Nudging as a Governance Tool

In behavioral economics and psychology, a **nudge** refers to a subtle cue or indirect suggestion that influences how people behave or make decisions—**without coercing them or limiting their choices**. The effectiveness of nudging lies in its gentle influence: it respects individual freedom while encouraging better choices through thoughtful design, context, and feedback.



image by liudmila chernetska from istock

9.1.1: What are Nudges?

Nudges are frequently utilized in areas such as public policy, marketing, and workplace culture to assist individuals in making decisions that lead to better outcomes—both for themselves and for the system as a whole. Common examples include:

- Placing healthy food at eye level in cafeterias to encourage healthier eating habits.

- **Setting default options** (e.g., opt-out organ donation) to promote socially beneficial behavior.
- **Sending reminders** or prompts to encourage timely actions, such as saving for retirement or updating software.

This concept gained global attention through the book *Nudge: Improving Decisions About Health, Wealth, and Happiness* by Richard Thaler and Cass Sunstein. They also introduced the idea of choice architecture—how the structure and presentation of choices influence decision-making. By designing the environments in which decisions are made, we can gently guide behavior toward better outcomes, often overcoming common cognitive biases.

9.1.2: Nudging in IT Architecture

In the realm of IT architecture, nudging serves as a powerful, non-intrusive method to guide teams and individuals toward more consistent, efficient, and aligned behavior—**without enforcing strict compliance**.

Here are some ways nudging manifests in architectural practice:

- **Architectural principles:** Serve as soft guidelines rather than hard rules, subtly encouraging teams to align with broader strategic goals.
- **Best practice recommendations:** Promote reuse, consistency, and improved quality across teams while still allowing for autonomy.
- **Golden paths:** Predefined, well-supported solutions or workflows that are the easiest to follow, gently guiding teams toward preferred technology choices. ([Spotify's golden paths](https://engineering.spotify.com/2020/08/how-we-use-golden-paths-to-solve-fragmentation-in-our-software-ecosystem/)¹ are a strong example.)
- **Dashboards and visualizations:** Lightweight analytics platforms can highlight outdated or poor-quality software, applying social and reputational pressure for improvement without direct enforcement.
- **Technical debt tracking:** Making technical debt visible raises awareness and nudges teams to address it over time.
- **Cloud cost visualizations:** Demonstrating cost trends per team encourages behavior that leads to greater efficiency in cloud usage.

¹<https://engineering.spotify.com/2020/08/how-we-use-golden-paths-to-solve-fragmentation-in-our-software-ecosystem/>

These nudges are most effective when they are **transparent, actionable, and connected to meaningful outcomes**. They constructively influence behavior—helping people and teams make smarter decisions **without the burden of top-down mandates** or punitive enforcement.

The principles of **Grounded Architecture** closely align with the concept of nudging. Instead of imposing rigid control, it creates environments that organically and collaboratively guide behavior. For instance:

- **Lightweight Architectural Analytics** tools are designed to highlight areas that need improvement—acting as cues rather than commands.
- **Collaborative Networks** encourage peer learning and the dissemination of best practices through community influence and storytelling.
- The **Operating Model** promotes autonomous decision-making while integrating subtle mechanisms that **nudge alignment** with strategic goals.

Together, these approaches create a governance strategy that is **flexible, respects autonomy**, and is **highly effective** in complex, dynamic organizational settings.

9.2: Taxation (Economic Incentives) as a Governance Mechanism

Taxation as governance is an approach that promotes accountability without imposing restrictions. Instead of prohibiting specific actions, it allows teams or departments to make their own decisions—but with a **cost**. This “cost,” or tax, is typically linked to the resources they consume. It creates a **feedback loop** that helps regulate behavior by making the consequences of excessive usage visible and tangible.

A common and effective example of this is found in **managing cloud costs**. In many organizations, departments consume public cloud services, such as storage, computation, and bandwidth. When these expenses are **cross-charged**—meaning each department pays for its share of usage—it sends a **clear signal about resource consumption**. Teams become more aware of their costs and are motivated to avoid waste and optimize efficiency.



image by steve buissinne from pixabay

This approach contrasts with **nudging**, which uses subtle design cues and informational guidance. Taxation, on the other hand, introduces **direct, measurable consequences**. For example, if a project consistently

exceeds its allocated cloud budget, it may be **delayed, reviewed, or even canceled**. This adds a layer of accountability to resource use while still preserving team autonomy.

9.2.1: The Role of Architecture in a Taxation Model

The architecture practice plays a **critical role** in supporting this governance approach by ensuring that **taxation mechanisms are fair, transparent, and data-driven**. Key elements include:

- **Transparent reporting:** All stakeholders must clearly understand how resource consumption is measured, how costs are allocated, and how these factors impact decision-making. This transparency builds trust and supports behavior change.
- **Accurate, comprehensive data:** Taxation should be based on actual usage patterns, not estimates. Detailed cost reports—especially from public cloud environments—form the basis of meaningful economic incentives.
- **Timely feedback loops:** Teams need **real-time or near-real-time insights** into their resource usage and associated costs. When feedback is delayed or unclear, the effectiveness of the tax diminishes.
- **Lightweight Architectural Analytics:** Tools like [Lightweight Architectural Analytics](#) aggregate cloud cost data, visualize spending patterns, and help identify optimization opportunities. These analytics transform raw data into **actionable insights**.
- **Collaborative Networks:** [Collaborative Networks](#) align teams around shared goals, support the interpretation of analytics data, and facilitate the exchange of ideas for optimizing usage. They help foster a **culture of responsible resource management**.

9.2.2: Driving Sustainable Change

Ultimately, governing with taxes enables organizations to balance **autonomy with accountability**. It allows teams the freedom to make choices while making the **impact of those choices visible** through economic signals. When implemented effectively, this approach:

- Encourages **cost-conscious decision-making**.
- Discourages **wasteful or unsustainable use** of shared resources.
- Promotes **optimization and innovation** within constraints.
- Strengthens alignment between **financial accountability** and **technical architecture**.

With the right tools and structures in place—such as Lightweight Architectural Analytics and Collaborative Networks—**architecture practices become essential enablers** of this governance style. They ensure that economic incentives are used wisely, feedback is timely, and resource consumption aligns with the organization’s long-term goals.

9.3: Mandates and Bans as Governance Tools

Mandates and bans are the most direct forms of governance. They clearly specify what people **must do** (mandates) or **must not do** (bans). These tools are essential for defining **non-negotiable rules and boundaries**, particularly in areas where compliance, ethics, or risk mitigation are crucial.

In practice, mandates and bans play a **limited but vital role** in shaping behavior. For example:

- A company may restrict the use of certain cloud providers to ensure data sovereignty or contractual compliance.
- Legal and regulatory mandates may require strong encryption, privacy protection, or specific financial reporting standards.
- Specific bans may prohibit risky technologies or architectural patterns that have proven unsustainable or insecure.



image by tumisu from pixabay

While mandates and bans can be powerful, they should be used **sparingly and intentionally**. Overusing them may create unnecessary friction, discourage innovation, or diminish trust. However, in areas like **security, compliance, and ethics**, clear rules are not just helpful—they are essential. These rules help prevent incidents, safeguard the organization’s reputation, and ensure adherence to laws and regulations.

9.3.1: The Role of Architecture in Enforcing Mandates and Bans

In this governance model, the architecture practice serves as a **contributor—not the sole authority**. Mandates and bans are typically shaped collaboratively with:

- **Security and risk teams** to address vulnerabilities and ensure compliance.
- **Legal and regulatory departments** to adhere to external obligations.
- **Operations and finance** to enforce constraints related to costs or performance.

The architecture team’s role is to **clarify, document, and reinforce** these mandates through guidance, tooling, and visibility:

- **Lightweight Architectural Analytics:**
[Lightweight Architectural Analytics](#) helps identify areas that require monitoring or enforcement. It can surface security vulnerabilities, policy violations, or infrastructure risks through dashboards and reports—creating **transparency and focus**.
- **Collaborative Networks:**
[Collaborative Networks](#) are essential for the **socialization and adoption** of mandates. They help explain the reasoning behind new rules, build consensus, and foster acceptance. These networks ensure that changes are **well-communicated, understood, and internalized**—not just enforced.

9.3.2: When and How to Use Mandates and Bans

Mandates and bans are best reserved for:

- **Critical compliance needs** (e.g., GDPR, financial regulations)
- **Organizational priorities** (e.g., sustainability, cybersecurity)
- **Strategic constraints** (e.g., vendor lock-in, architectural standards)

They work most effectively when:

- Stakeholders are **engaged early in the process**.
- There is **clear, data-driven justification**.
- Communication is **open, empathetic, and thorough**.
- There are **mechanisms to track adherence and measure impact**.

Governing with mandates and bans involves setting **firm, transparent boundaries**, but this should be done with care. This governance tool is best employed when other methods (like nudging or taxation) are insufficient to ensure compliance, protect the organization, or uphold its values.

When thoughtfully implemented, and supported by clear analytics and strong collaboration, mandates and bans can become **protective measures**, not roadblocks—enabling safe innovation within a framework of trust and responsibility.

9.4: Questions to Consider

- *What are the key components of the governance model in your organization, and how do mandates, taxes, and nudging influence them?*
- *How does your organization currently handle mandates and bans? Are they explicit and aligned with the overall technology strategy?*
- *How effective is the enforcement of these mandates and bans in your organization? Could improvements be made to create clarity and provide transparency?*
- *How does your organization approach taxation as a form of governance? Is it transparent, data-driven, and efficient?*
- *Can you identify any examples of ‘nudging’ in your current architectural environment? How effective are these subtle suggestions in influencing behavior or decision-making?*
- *How does your organization promote best practices and align around them? Are there any ‘golden paths’ for technology choices?*
- *How are your organization’s tech debt and the cost trends of cloud services tracked and visualized? Do these methods create enough awareness to stimulate improvement?*
- *How could you better utilize nudging to improve organizational decision-making? What biases or barriers to effective decision-making could you target with this approach?*

10: Transforming Organizations with Grounded Architecture



image by gremlin from istock

IN THIS SECTION, YOU WILL: Understand the value that an architecture practice based on the ideas of Grounded Architecture can create for an organization.

KEY POINTS:

- When a Grounded Architecture framework is in place, it can positively transform an organization's functioning.
- These impact categories are Executing At Scale, Improving the Quality of Decision-Making with Data, Maximizing Organizational Alignment & Learning, and Higher Adaptivity.

When a Grounded Architecture framework is in place, it can positively transform an organization's functioning. These categories of impact, aligned with [defined goals](#), are:

- Enabling Execution of Architecture Practice At Scale,
- Increasing Architecture Practice Adaptivity,
- Improving the Quality of Decision-Making with Data,
- Maximizing Organizational Alignment, and
- Maximizing Organizational Learning.

10.1: Executing at Scale

Our first goal was to find a way to support hundreds of teams and thousands of projects with significant complexity and diversity.

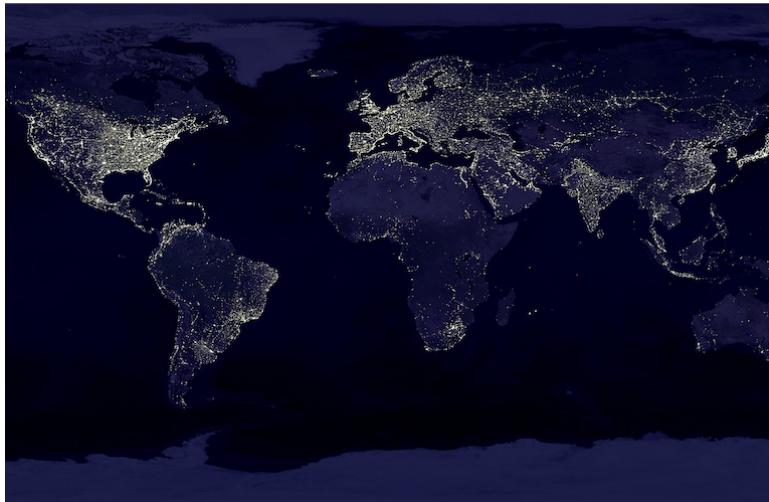


image by wikiimages from pixabay

Each element of Grounded Architecture enables an architecture practice to operate at scale in several ways.

10.1.1: Lightweight Architectural Analytics

The **Lightweight Architectural Analytics** can support large-scale operations by:

- **Self-Service Data Access:** Providing data as a self-service resource allows teams to access the information they need independently, reducing the reliance on manual sharing methods. This service can

be facilitated through internal websites or portals where data tools and insights are readily available.

- **Elimination of Meetings:** As architects frequently own some unique pieces of information, in particular related to the big picture and overall dependencies, lots of architecture support goes on providing that information to the teams. Typically that means having meetings, workshops and exchanging lots of messages. With Lightweight Architectural Analytics, we significantly reduced the need for such information-sharing or data-gathering meetings (every time we added a new data app, I got a few hours back in my calendar and exchanged fewer messages). Sharing data via Architectural Analytics dashboards also generates a useful feedback about how data are used, and helps ensuring that data is up-to-date and accessible.
- **Automation:** Automation minimizes manual effort, which is crucial as manual processes do not scale efficiently. Automating data management processes ensures that data can be collected, processed, and analyzed without extensive human intervention.

10.1.2: Collaborative Networks

The Collaborative Networks enhances execution at scale by focusing on:

- **Developing Connections:** Building strong relationships at all levels of the organization is crucial. This network facilitates quicker alignment of objectives, efficient information sharing, and swift execution of shared decisions.
- **Speeding Up Alignment:** Effective communication channels and collaborative tools help align teams rapidly, ensuring everyone is on the same page and working towards common goals.
- **Facilitating Shared Decisions:** Enabling a culture where shared decisions are made quickly can enhance the responsiveness and adaptability of the organization.

10.1.3: Operating Model

The Operating Model promotes execution at scale by:

- **Distributed Decision-Making:** Distributing decision-making across an organization prevents bottlenecks associated with centralized decision-making processes. This model empowers more people to take ownership and make decisions within their scope, leading to faster and more effective outcomes.
- **Promoting a Collaborative Operating Model:** An operating model that supports distributed decision-making and collaboration ensures that the organization can handle more projects simultaneously without overburdening any single entity or individual.

By leveraging these foundations, we managed the complexities and diversity of numerous projects and teams more efficiently, ensuring scalability and effective execution.

10.2: Adaptivity

The second goal, ensuring that an architecture practice can adapt quickly to stay relevant in new contexts, is crucial for maintaining an organization's agility and resilience.

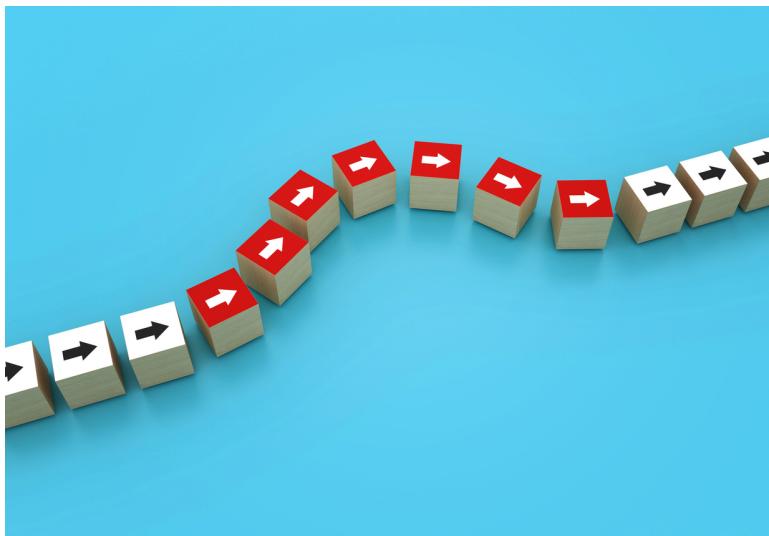


image by porcorex from istock

The Grounded Architecture framework has core elements that provide a highly flexible and adaptive setting. Here are the key drivers of flexibility within this structure:

10.2.1: Lightweight Architectural Analytics

The Lightweight Architectural Analytics can support adaptability by:

- **Automation and Extensibility:** We used lean and flexible tools that we could swiftly extend and reconfigure to accommodate changes. For example, following acquisitions or mergers, with our lean tools and automation scripts, we were able to integrate all new

source code repositories and many other data sources in a matter of hours or days. Automation ensures that data remains relevant and up-to-date, providing essential connections and feedback tailored to the organization's evolving needs.

- **Configurable Adaptation to Organizational Changes:** We created robust Lightweight Architectural Analytics to adapt its views to the changing realities of different parts of the organization, ensuring that data insights are always aligned with current operational contexts. For instance, we developed tools that enable explorative, on-the-fly aggregation of diverse data sources (e.g., created aggregated source code statistic reports based on repository tags or name conventions).

10.2.2: Collaborative Networks

The Collaborative Networks can support adaptability by:

- **Capacity Redistribution:** Collaborative Networks, through a central team, can alleviate the temporary capacity shortages experienced by distributed teams. This flexibility ensures that an architecture practice continue to operate smoothly even during periods of high demand or limited resources.
- **Decentralized Support:** By maintaining a network of well-connected architects, an architecture practice can support the organization effectively without relying solely on a central team. This decentralization fosters resilience and adaptability, enabling architectural guidance and oversight across various teams and projects.

10.2.3: Operating Model

The Operating Model can support adaptability by:

- **Flexible Decision-Making:** The Operating Model supports adaptivity by promoting a flexible setting and distributing decision-making authority throughout the company. This distribution helps prevent an architecture practice from becoming a bottleneck or a single point of failure.

- **Delegation of Architectural Decisions:** Senior architects can delegate most architectural decisions to teams, allowing them to focus on critical strategic initiatives, such as defining cloud, data, or platform strategies and supporting significant decisions related to mergers and acquisitions. By grounding the architecture with data and people connections, organizations empower senior architects to concentrate on high-impact areas.

This structure enhances an architecture practice's flexibility and adaptivity. It ensured we could respond promptly and effectively to new challenges and opportunities.

10.3: Improving the Quality of Decision-Making with Data

The third goal stated that we need tools and mechanisms to make a decision process more data-informed and less dependent on opinions. There are significant benefits to making our decision process as much as possible data-driven. Architectural discussions can be heated and opinionated, not leading to the best arguments and decisions.



image by cofotoisme from istock

To make the decision process more data-informed and less dependent on opinions, we need to focus on the interplay between three foundational elements. These elements collectively create a robust framework for data-driven decision-making in architectural discussions.

10.3.1: Lightweight Architectural Analytics

The **Lightweight Architectural Analytics** ensures the availability and readiness of data needed for decisions, fueling data-informed discussions:

- **Data Collection:** Systematically gather high-quality data on relevant internal and external technology developments.

- **Data Management:** Maintain a well-organized and easily accessible database that supports quickly retrieving relevant data.
- **Data Analytics:** Employ analytics tools to process and interpret the data, providing meaningful insights to support decisions.
- **Real-Time Data Availability:** Ensure that data is updated regularly and available in real-time to inform ongoing discussions and decisions.

10.3.2: Collaborative Networks

Data is not enough; you need to make this data available to decision-makers. The **Collaborative Networks** ensures people are well-connected to share information and make decisions via:

- **Expert Network:** Establish and maintain a network of experts and stakeholders who can provide insights on various architectural aspects.
- **Collaboration Tools:** Implement collaboration tools that facilitate seamless communication and information sharing among team members.
- **Training and Development:** Provide training programs to enhance the data literacy and analytical skills of architects and decision-makers.
- **Engagement Practices:** Develop practices for regularly engaging stakeholders and ensuring their input is considered in decision-making.

10.3.3: Operating Model

The **Operating Model** provides processes that enable architects to move from opinion-based decisions to data-driven economic risk modeling via:

- **Process Implementation:** Implement processes that guide architects in dismantling hype and buzzwords, presenting problems clearly, and making data-driven decisions.

- **Data Integration:** Develop methodologies for integrating relevant data into discussions, ensuring that data supports and guides the conversation.
- **Economic Risk Modeling:** Create models that translate drivers and data into economic risk assessments, helping to identify the best solutions for the given business context.
- **Decision Support Tools:** Deploy tools and technologies that assist in visualizing data, modeling risks, and evaluating options.

Focusing on these foundational elements and their key actions can transform our architectural discussions into a more data-informed process. This focus ultimately led us to better, more objective decision-making that aligns with our business context and goals.

10.4: Maximizing Organizational Alignment

The fourth goal emphasizes that an architecture practice should be a cohesive factor in minimizing misalignments within large organizations.



image by 36clicks from istock

Misalignments often occur in such settings due to various factors, including complex structures and diverse objectives. However, a well-grounded architecture can address and reduce these misalignments through its foundations.

10.4.1: Lightweight Architectural Analytics

The **Lightweight Architectural Analytics** improves organization alignment by **creating transparency**. By establishing a robust Lightweight Architectural Analytics, organizations can enhance transparency, which is necessary for building trust and facilitating alignment. This involves making data easily accessible and understandable across different departments, ensuring everyone has the same information and can make

informed decisions. Transparent data practices help align objectives and actions, reducing the chances of misalignment.

10.4.2: Collaborative Networks

The **Collaborative Networks** improves organization alignment by **facilitating collaboration**. Collaborative Networks focuses on developing global structures that connect employees across various functions and geographies. By making it easier for people to collaborate, share knowledge, and work together, this foundation helps create a unified approach to organizational goals. Collaborative environments foster alignment by ensuring that everyone is working towards common objectives.

10.4.3: Operating Model

The **Operating Model** improves organization alignment by:

- Facilitating **pre-decision alignment** via **collaborative decision-making**. This platform plays a pivotal role in minimizing misalignments by enabling individuals and teams working on similar projects or topics to identify each other and collaborate. This early alignment minimizes duplication of efforts and optimizes resource utilization, ensuring that efforts are aligned from the outset.
- Facilitating **post-decision dissemination** via **knowledge sharing and more awareness**. After decisions are made, we can leverage the collaborative networks to ensure that these decisions are communicated across the organization. This widespread dissemination helps all parts of the organization benefit from the insights and lessons learned from one unit, leading to more cohesive and aligned operations.

An architecture practice can reduce misalignments and drive a more unified, efficient, and aligned organization by integrating data transparency, enhanced collaboration, and structured processes for decision-making and dissemination.

10.5: Maximizing Organizational Learning

“Good judgment comes from experience, and experience comes from bad judgment.” –Fred Brooks

Our last goal is that architecture should help organizations learn quickly, stay up-to-date with emerging technologies and industry trends, and recommend technology upgrades. Learning is one of the primary daily tasks of architects. Architects must proactively identify relevant new technology developments and create pragmatic technology recommendations for concrete platforms across the organization based on their understanding.



image by rawpixel from istock

The Grounded Architecture framework offers comprehensive support through its Lightweight Architectural Analytics, Collaborative Networks, and Operating Model to help organizations learn quickly, stay up-to-date with emerging technologies, and recommend technology upgrades.

10.5.1: Lightweight Architectural Analytics

Lightweight Architectural Analytics plays a pivotal role in accelerating the learning and adoption of new technologies. Here's how:

- **Facilitating Exploration and Reflection:** Lightweight Architectural Analytics enables individuals and teams to explore new tools and technologies effectively by providing relevant and comprehensive data. Access to this data allows for thorough experimentation, analysis of outcomes, and reflection, which is crucial for understanding and refining new technology implementations. For instance, it may not be easy to calculate the cost of a new cloud service. However, running proofs-of-concept and analyzing collected data can provide good insights.
- **Supporting Informed Decision-Making:** Architects and teams can make informed decisions regarding technology adoption and upgrades with up-to-date data. This minimizes the risk associated with implementing new technologies and ensures that the choices are based on solid evidence and insights.

10.5.2: Collaborative Networks

Collaborative Networks enhance learning by creating and maintaining a culture of knowledge sharing. It achieves this through:

- **Spaces for Sharing Knowledge:** Regular update calls, knowledge-sharing sessions, and conferences need to be organized to facilitate the exchange of architectural and technological knowledge. These events are critical for informing the organization about the latest developments and best practices.
- **Maximizing Personal Learning:** Collaborative Networks ensure that individual lessons are transformed into shared guidelines by deriving generalized insights from cross-group cases. This collective intelligence benefits the entire organization and fosters continuous personal and professional growth among team members.

10.5.3: Operating Model

The Operating Model integrates learning into daily workflows through structured processes:

- **Embedding Learning into Processes:** Learning is accelerated by embedding it into the organization's processes. This is achieved by defining and distributing knowledge-sharing and lesson-learned processes across the organization. By doing so, learning becomes a seamless part of daily activities rather than an additional task.
- **Systematic Knowledge Capture and Application:** The platform ensures that knowledge is systematically captured, shared, and applied. This approach minimizes overhead while maximizing learning opportunities, enabling the organization to quickly adapt and apply new insights and technologies.

By leveraging Lightweight Architectural Analytics, Collaborative Networks, and Operating Model, the Grounded Architecture framework ensures that learning is continuous, efficient, and embedded within the organization's fabric. This comprehensive support system helps organizations stay up-to-date with emerging technologies and industry trends and fosters a proactive learning environment that drives innovation and growth.

10.6: Questions to Consider

It is always essential to be thoughtful about the value and impact of your work. Ask yourself the following questions:

- *How effective is your organization's current an architecture practice at scale? How valuable could principles of Grounded Architecture be in enhancing its efficiency?*
- *To what extent does your organization use data to inform architectural decisions? What steps could you take to move your organization from opinion-based to more data-driven decision-making?*
- *How well-aligned are the different areas within your organization, and how does this affect your an architecture practice? Could the Lightweight Architectural Analytics and Collaborative Networks principles be utilized to improve alignment?*
- *What strategies does your organization currently have to foster organizational learning? How could the methods described in the Grounded Architecture model enhance this?*
- *How quickly can your organization adopt and utilize new technologies? How could your an architecture practice accelerate this process?*
- *Consider the adaptivity of your organization's an architecture practice. How could your an architecture practice improve it?*
- *Reflecting on the value of the "Lightweight Architectural Analytics" concept, how effectively is your organization tracking changes or supporting what-if scenarios analysis?*
- *What role do most senior architects play in your organization? Could their time be better utilized on strategic initiatives?*
- *How sustainable is an architecture practice in your organization in the absence of a strong central team? Could implementing a Lightweight Architectural Analytics and well-connected architects help mitigate this?*

11: Next Steps: Leveraging The Potential of Generative AI

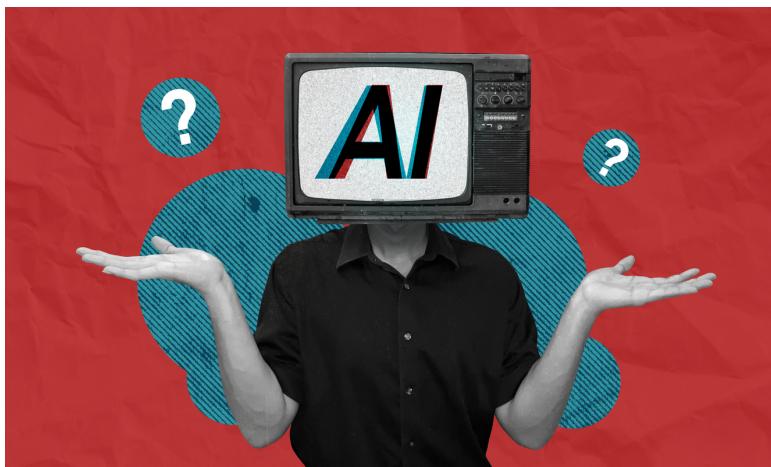


image by mininyx doodle from istock

IN THIS SECTION, YOU WILL: Learn how you can responsibly leverage Generative AI as a powerful augmentation tool within to enhance efficiency, data-driven insights, and collaboration, provided you proactively manage its inherent risks and maintain critical human oversight.

KEY POINTS:

- Generative AI (GenAI) offers significant potential to augment your work as an IT architect by enhancing data analysis (Lightweight Analytics), improving knowledge sharing (Collaborative Networks), and streamlining tasks (Operating Model).
- Practical applications include accelerating analytics, generating ADRs and diagrams, augmenting requirements analysis, assisting solution design, improving code reviews, and drafting communications, ultimately freeing you up for higher-value strategic thinking.
- While GenAI promises increased efficiency, consistency, and better data-driven decisions, you must navigate challenges like accuracy issues (hallucinations), security risks, ethical considerations (bias), and the need for continuous human oversight and judgment.
- Responsible adoption requires a principled approach focusing on clear objectives, strong data governance, human-in-the-loop validation, grounding AI with RAG, and fostering an AI-literate culture within your organization.
- The future points towards real-time, augmented EA, agentic AI, and DTOs, further emphasizing the need for adaptability and evolving your role towards governing AI and focusing on strategic, collaborative, and ethical considerations.

This section explores the exciting intersection of Generative AI (GenAI) with IT Architecture, particularly through the lens of the Grounded Architecture framework.

Generative AI models are sophisticated tools that can create new content like text, computer code, images, and even complex designs. They are rapidly moving from experimental stages to practical, real-world applications, transforming industries and how businesses operate. The potential for GenAI to significantly improve IT and Enterprise Architecture (EA) is especially strong. Traditionally, architects have struggled to manage rapidly changing business needs with tools that are often slow, disconnected, and static. GenAI offers a chance to make architecture work more effective, automate repetitive tasks, and enable faster, more

informed decisions based on data. This ushers in a new era of efficiency and innovation in the field.

However, using GenAI without a clear strategy could create more complexity or new problems. The Grounded Architecture approach provides a solid foundation for harnessing GenAI's power while managing its risks. This section serves as a comprehensive and practical guide for IT and Enterprise Architects looking to integrate GenAI into their Grounded Architecture practices. We will cover GenAI's capabilities, key ways to integrate it, common uses, potential benefits and challenges, best practices for responsible use, and what's on the horizon. Our aim is to empower architects to use GenAI not just as another tool, but as a **strategic asset within their architectural work**.

In [the appendix¹](#), you'll find example Generative AI prompts I used in practice for many of the tasks we will discuss in this section.

(To “eat my own dog food,” I drafted this report with initial research and insights from the Gemini Deep Research chatbot. I then refined the narrative and coherence using ChatGPT, a language generation model, and checked language and grammar with Grammarly. Finally, I manually edited it extensively to ensure accuracy and relevance.)

¹[gen-ai-prompts](#)

11.1: What Generative AI Can Do for Today's Architect

Generative AI, or GenAI, refers to artificial intelligence systems trained on vast amounts of data to create new, realistic content—such as text, code, images, and designs—without simply copying the original data. Unlike traditional AI, which mainly analyzes data and makes predictions, GenAI is specifically **designed for creation**. Users typically interact with these models using plain language commands (prompts), unlocking a variety of capabilities highly relevant to IT and Enterprise Architects.



image by lemon tm from istock

GenAI provides architects with powerful tools for creating, analyzing, automating, communicating, and making informed decisions:

- **Creating and Improving Content:** Quickly draft reports, technical documents, emails, meeting summaries, key decision logs (known as Architecture Decision Records or ADRs), code snippets, and initial architecture diagrams. Large Language Models (LLMs), a type of GenAI, are central to this.
- **Analyzing Data and Spotting Patterns:** Examine large volumes of information, both structured (like databases) and unstructured

(like text documents), to find common issues in architecture (anti-patterns), security weaknesses, and outdated technology (technical debt). GenAI can also convert architectural diagrams into structured data for easier analysis.

- **Automating Tasks for Greater Efficiency:** Automate tasks like creating documentation, performing routine code reviews, generating standard reports, extracting requirements from documents, and brainstorming early solution ideas. This frees up architects to focus on more complex challenges.
- **Improving Interaction and Communication:** Power advanced chatbots and conversational assistants that provide instant access to company knowledge, answer questions about architecture, explain complex technical ideas simply, and share insights across the organization.
- **Offering Recommendations and Suggestions:** Propose potential solutions, recommend suitable technologies or architectural patterns, suggest optimizations, and outline transition plans. However, the final validation and decision-making always remain with the architects.

A critical technology for using GenAI in businesses is **Retrieval-Augmented Generation (RAG)**. Standard GenAI models generate responses based only on the data they were initially trained on. RAG improves this by first searching for and retrieving relevant, up-to-date information from trusted company-specific sources—like internal documents, databases, or Enterprise Architecture repositories—before creating a response. This ensures that the AI’s output is based on current and reliable company data.

RAG is vital for **making GenAI outputs trustworthy**, accurate, and relevant to the specific business context. It helps reduce AI “hallucinations” (incorrect or nonsensical outputs), ensures decisions are based on verified knowledge, and strengthens the data-driven foundation of Grounded Architecture.

11.2: Integrating GenAI with Grounded Architecture: Connecting Capabilities to the Framework

Generative AI can enhance the architecture practice within the Grounded Architecture framework by improving data analysis, knowledge sharing, and operational efficiency. These enhancements can make the framework more scalable, dynamic, and impactful (Figure 1).

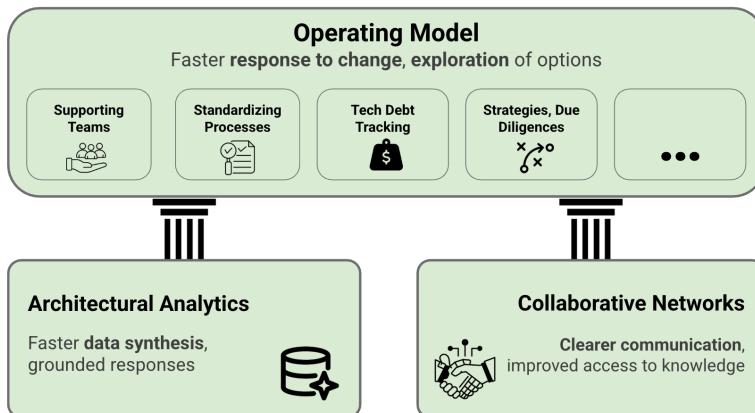


Figure 1: An overview of potential Generative AI potential in the context of the Grounded Architecture Framework.

By integrating GenAI capabilities into each part of Grounded Architecture, significant new value can be created:

- GenAI in Lightweight Architectural Analytics (Understanding the Current State)
 - Gathering and Processing Data: Automate the collection, interpretation, and summarization of information from various sources like code repositories, cloud service bills, customer support tickets, and existing documentation. RAG can help query and consolidate scattered information.

- **Recognizing Patterns and Anomalies:** Improve the detection of architectural patterns, common pitfalls (anti-patterns), security vulnerabilities, outdated technologies, and accumulated technical debt using advanced GenAI analysis.
- **Generating Reports and Dashboards:** Automatically create architectural reports and visual dashboards from the analyzed data, making them available more quickly and consistently.

- **GenAI in Collaborative Networks (Working Together)**

- **Managing and Sharing Knowledge:** Transform static information stores, such as documents and Architecture Decision Records (ADRs), into dynamic, easily searchable knowledge bases using RAG-powered conversational interfaces (like chatbots).
- **Assisting with Communication:** Help draft clear communications tailored to specific audiences, whether they are technical colleagues or business executives, including emails, summaries, and explanations.
- **Summarizing Meetings:** Automatically generate summaries of meetings, capturing key decisions and action items from recordings or transcripts.

- **GenAI in the Operating Model (How Architecture Gets Done)**

- **Supporting Coding and Documentation:** Provide AI-powered assistance for writing code, creating technical documentation, and analyzing project requirements.
- **Generating Architectural Artifacts:** Automate or assist in creating architecture diagrams, Architecture Decision Records (ADRs), compliance documents, and operational reports.
- **Tracking Technical Debt:** Improve tools for analyzing technical debt by summarizing findings, identifying patterns, and prioritizing fixes based on their business impact. This includes flagging aging technologies.
- **Speeding Up Due Diligence:** Accelerate the review of technical documents during company mergers, acquisitions, or when evaluating new technologies by using GenAI for summarization.

- **Standardizing Processes:** Help draft standards, policies, governance procedures, and ADRs. AI agents can also check if proposed changes comply with these standards and maintain consistency.
- **Developing Strategy:** Assist in creating initial drafts of strategy documents (for areas like cloud adoption, data management, or technology platforms) by summarizing the current situation based on analytics and suggesting options for the future.

GenAI streamlines architecture work by automating analysis, improving communication, and accelerating strategy and decision-making across the Lightweight Architectural Analytics, Collaborative Networks, and Operating Model pillars.

11.3: Practical Examples: GenAI in Action

GenAI can significantly **help architects** by taking over repetitive tasks. This allows them to **focus on more strategic thinking**, weighing complex options, collaborating effectively, interpreting results, and applying human judgment—all core activities in the Grounded Architecture approach.



image by blackjack3d from istock

11.3.1: Use Case 1: Analysing Dependencies and Technical Debt

- **Scenario:** You need to assess how different microservices depend on each other and identify technical debt (like unnecessary connections or outdated software libraries).
- **GenAI Application:** Use GenAI to automatically parse and summarize data from code repositories, software deployment pipelines (CI/CD), and application performance monitoring (APM) tools. Employ RAG to search internal documentation for important context (like who owns what, or relevant company standards). AI agents or specialized technical debt tools can then identify problem patterns and aging technologies. Finally, GenAI can generate a draft report of these findings.

- **Grounded Architecture Link:** Directly accelerates Lightweight Architectural Analytics.

11.3.2: Use Case 2: Generating Architecture Decision Records (ADRs)

- **Scenario:** After a team discussion, you need to document a decision about which messaging queue technology to use.
- **GenAI Application:** GenAI can summarize meeting notes or recordings. You provide this summary and an ADR template to GenAI. Using RAG, the AI can access relevant context (like project requirements or related past decisions). GenAI then drafts key sections of the ADR (like the problem, the decision made, the reasons why, and the expected outcomes), which the architect reviews and finalizes.
- **Grounded Architecture Link:** Supports the Operating Model (by standardizing documentation) and Collaborative Networks (by capturing knowledge).

11.3.3: Use Case 3: Creating Architecture Diagrams from Descriptions

- **Scenario:** You need to quickly create a system overview diagram (like a C4 context or component diagram) for a discussion.
- **GenAI Application:** Use an AI-powered diagramming tool (e.g., Eraser.io, Diagramming AI). You describe the system, its parts, and how they connect using plain language. The tool generates an initial diagram, which you can then refine with more instructions or manual adjustments. Some tools can also create diagrams from code (Infrastructure as Code) or sketches.
- **Grounded Architecture Link:** Speeds up the creation of visuals for Collaborative Networks and supports design activities in the Operating Model.

11.3.4: Use Case 4: Improving Requirements Analysis

- **Scenario:** You have various inputs like user stories, emails, and meeting transcripts, and you need to combine them into a structured list of requirements.
- **GenAI Application:** Natural Language Processing (NLP) tools (a type of GenAI) can process these inputs to extract key requirements and user needs, summarize long documents, identify unclear statements or conflicts, help structure the information, and even draft initial acceptance criteria and test scenarios.
- **Grounded Architecture Link:** Supports the early design phase within the Operating Model, ensuring new solutions meet stated requirements.

11.3.5: Use Case 5: Assisting with Solution Design and Evaluation

- **Scenario:** You need to explore different architectural strategies for a new recommendation engine and evaluate them based on quality goals (like performance or security).
- **GenAI Application:** Use GenAI with RAG (giving it access to internal standards and data) to suggest relevant design patterns. It can generate initial descriptions and basic code structures. GenAI can also help evaluate options by summarizing technical documentation, performing competitor analysis, and finding performance benchmarks. Some advanced tools can even simulate how different designs might perform.
- **Grounded Architecture Link:** Supports design and strategy work within the Operating Model, using data and established patterns to guide decisions.

11.3.6: Use Case 6: Enhancing Code Review Processes

- **Scenario:** You need to review new code to ensure it meets company standards, identify potential bugs, check for security issues, and

assess if it adds to technical debt.

- **GenAI Application:** Integrate AI-powered code review tools into your development pipeline (CI/CD process). These tools can scan code changes for style violations, errors, overly complex sections, and known bad practices. They can also identify security vulnerabilities, generate summaries for review, and sometimes suggest fixes.
- **Grounded Architecture Link:** Supports the Operating Model by improving code quality, enforcing standards, and reducing technical debt.

11.3.7: Use Case 7: Drafting Communications for Stakeholders

- **Scenario:** You need to explain a technical decision (like migrating to a new cloud platform) to business executives who may not have a technical background.
- **GenAI Application:** Provide the technical reasons and supporting data to GenAI. Instruct the AI to draft an executive summary or email in clear, business-friendly language, avoiding jargon. It can also effectively summarize complex reports for different audiences.
- **Grounded Architecture Link:** Makes Collaborative Networks more effective through clearer communication.

These use cases illustrate how Generative AI can enhance architectural practices by automating routine tasks and enabling architects to focus on strategic and high-value work. It helps streamline decision-making, improve documentation, enhance communication, and accelerate design processes, ultimately boosting efficiency and architectural quality.

11.4: Balancing Benefits, Risks, and Challenges

GenAI can significantly boost the core advantages of the Grounded Architecture framework. These benefits align well with the principles of Grounded Architecture: efficiency supports Pragmatism, faster data analysis underpins Data-Driven Decisions, better knowledge sharing enhances Collaborative Networks, and improved option evaluation fosters Adaptability:

- **Increased Efficiency and Productivity:** Automate and speed up tasks like drafting documents (e.g., Standard Operating Procedures, Architecture Decision Records), creating diagrams, analyzing data, and reviewing code. This saves valuable time, allowing architects to focus on strategic work.
- **Improved Consistency and Quality:** Ensure that all architectural documents and diagrams uniformly follow established standards and best practices. Consistent code reviews can reduce variations and minimize human error.
- **Faster Data-Driven Decision Support:** More quickly process, combine, and summarize diverse datasets for Lightweight Architectural Analytics. This leads to faster insights and the ability to uncover subtle patterns, supporting a core principle of Grounded Architecture.
- **Enhanced Collaboration and Knowledge Sharing:** Make the collective knowledge within Collaborative Networks easily accessible through RAG-powered chatbots and search tools. Clearer communication, aided by AI summarization and tailoring content for specific audiences, can also speed up onboarding for new team members.
- **Fostering Innovation:** By freeing up architects' time, GenAI creates more space for innovation. It can help explore different design options and generate new ideas based on data patterns.
- **Making Architectural Insights More Accessible:** Present architectural information to non-technical stakeholders through conversational interfaces and visualizations. This aligns with Grounded Architecture's goal of embedding architectural thinking throughout the organization.

However, benefits often come with risks (for example, gaining speed might sacrifice accuracy, or analyzing data might raise privacy concerns).

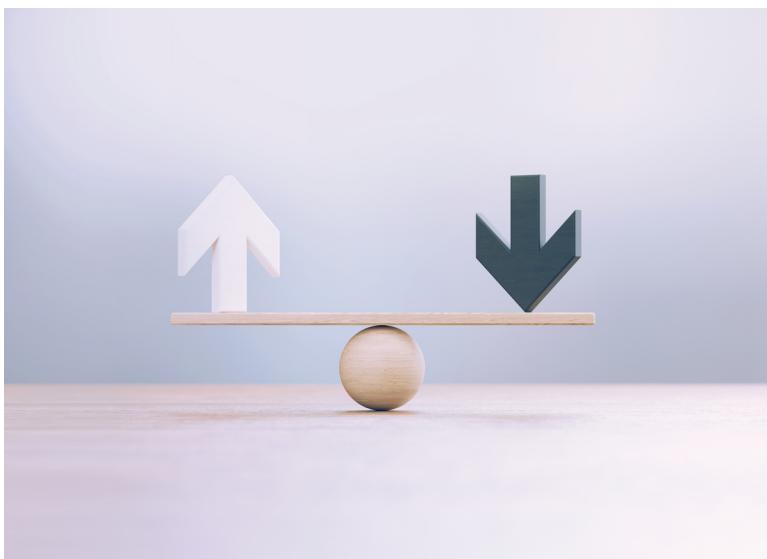


image by microstockhub from istock

To realize these benefits, it's crucial to actively manage these risks, emphasizing good governance, ethical considerations, and human oversight:

- **Accuracy and Reliability (AI “Hallucinations”):** GenAI can sometimes produce incorrect, nonsensical, biased, or completely fabricated information. Careful human review and validation are essential, which can use up some of the time saved by automation. Ensuring enterprise-level reliability is an ongoing challenge.
- **Security and Data Privacy:** Feeding confidential company data into GenAI models, especially those hosted on public cloud services, creates risks of data leaks, unauthorized access, or misuse of data for training other models. Strict access controls (like verifying user identity for every request rather than granting broad permissions), data encryption, and ensuring data stays within required geographical locations are vital.

- **Ethical Issues and Bias:** AI models can reflect and even amplify biases present in their training data, leading to unfair or problematic outcomes. It's important to proactively detect and reduce these biases.
- **Intellectual Property (IP) and Copyright:** The legal rules around ownership of AI-generated content are still developing. This raises potential risks of infringing on copyrights if the AI was trained on protected material. Often, there's a lack of clear guarantees regarding IP protection.
- **Need for Human Oversight and Judgment:** GenAI should assist architects, not replace their critical thinking, understanding of context, and strategic judgment. Over-relying on GenAI (automation bias) can lead to poor decisions; human expertise is still essential.
- **Cost and Resource Requirements:** Implementing GenAI can be expensive, requiring significant computing power (like GPUs) and specialized expertise. The ongoing costs of using GenAI (inference costs) can also be substantial.
- **Speed and Performance (Latency):** For applications that need real-time responses, GenAI might be too slow due to the complex calculations involved in generating content or analysis. This could affect user experience or hit usage limits.
- **Complexity of Integration:** Fitting GenAI into existing enterprise architecture tools and workflows is complex. It involves managing APIs (interfaces for software to talk to each other), data pipelines, designing effective prompts, setting up RAG context, and coordinating different components.
- **Model Limitations and Context Window:** GenAI models have a limit on how much information they can consider at once (their “context window”), which can be an issue for very large inputs. They may also struggle with truly new situations, complex reasoning, or generating genuinely innovative designs that go beyond existing patterns.
- **Dependence on Data Quality:** The reliability of AI, especially when using RAG, heavily depends on the quality, accuracy, consistency, and accessibility of the underlying data sources. Poor data management can lead to flawed AI outputs.

In conclusion, while GenAI offers significant potential to enhance architectural work, these benefits can only be fully realized by proactively managing its risks—ensuring robust governance, safeguarding data and ethics,

maintaining human oversight, and addressing technical and operational challenges with care and clarity.

11.5: Best Practices for Responsibly Adopting GenAI

The core principles of Grounded Architecture—being data-driven, fostering collaborative networks, and having a pragmatic operating model—provide a strong foundation for implementing these best practices.



image by aree sarak from istock

Organizations already following Grounded Architecture may find themselves well-prepared for responsible AI adoption.

- **Start with Clear Goals & Prioritized Uses:** Define specific, measurable objectives. Focus on uses that align with your strategy and Grounded Architecture principles. Begin with lower-risk experiments.
- **Establish Strong Data Governance:** Ensure your data is high-quality, secure, private, and consistently managed, especially for grounding AI/RAG systems. Implement robust policies for data quality, privacy, security, access control (using user identity), encryption, and data retention. Keep track of data origins (lineage).

- **Implement Human-in-the-Loop (HITL) & Oversight:** Design workflows that include points for human review and validation. Architects should always have the final say. Clearly define who is accountable. Guard against simply trusting AI outputs without scrutiny (automation bias).
- **Adopt a Principle-Based Governance Framework:** Use core ethical principles such as Fairness, Reliability, Safety, Privacy & Security, Inclusiveness, Transparency, and Accountability. Form an AI review board or a Center of Excellence (CoE). Document all decisions.
- **Focus on Grounding and Context (RAG):** For business use, prioritize RAG to ensure outputs are relevant and to minimize AI “hallucinations.” Connect GenAI to curated, reliable internal knowledge sources. Prepare your data so it can be easily retrieved by RAG systems.
- **Choose the Right Tools and Models (Build vs. Buy):** Make deliberate decisions about your implementation strategy. Evaluate different options: using AI features embedded in existing tools, using public GenAI model APIs, or building custom models. Consider techniques like fine-tuning, prompt engineering, AI agents, and RAG to customize models.
- **Architect for Security and Modularity:** Incorporate “Security by Design.” Limit system permissions and use individual user context for authorizing access. Build modular AI systems that are flexible and easier to manage for risk.
- **Test, Monitor, and Iterate:** Treat GenAI systems like products that need ongoing improvement. Conduct thorough pilot programs. Implement continuous monitoring for performance, accuracy, changes in behavior (drift), bias, and cost (this is sometimes called AI FinOps). Create feedback loops for continuous improvement.
- **Promote AI Literacy and a Culture of Responsible Use:** Train users on GenAI’s capabilities, limitations, risks, and ethical considerations. Encourage critical evaluation of AI outputs and open discussion about any concerns. Clearly communicate AI usage policies.

In conclusion, organizations grounded in data-driven practices, collaboration, and pragmatic operations are well-positioned to adopt GenAI responsibly—but success depends on setting clear goals, ensuring strong

governance, maintaining human oversight, and fostering a culture of continuous learning, ethical use, and thoughtful system design.

11.6: The Evolving Landscape

Grounded Architecture appears well-suited to take advantage of future AI trends. Its principles—being data-driven, adaptable, collaborative, and pragmatic—remain resilient and relevant. Its emphasis on data directly supports AI grounding; its collaborative networks facilitate ethical governance; and its adaptable operating model can incorporate new AI tools.



image by galeanu mihai from istock

Emerging AI trends suggest even faster feedback loops within the Grounded Architecture framework:

- Lightweight Architectural Analytics could become **nearly instantaneous** through AI monitoring.
- **Insights could be synthesized and shared rapidly** within Collaborative Networks.
- The Operating Model could be **more dynamic and adaptive** thanks to AI-driven analysis and recommendations.

These trends enhance the agility, responsiveness, and strategic value of Enterprise Architecture when guided by Grounded Architecture principles:

- **Towards Real-Time, Augmented Enterprise Architecture:** A shift from periodic documentation to a dynamic, “living” EA. AI agents could continuously monitor digital signals, updating architectural models and knowledge graphs automatically. Architects would become “augmented architects,” using AI as a “cognitive assistant” or “copilot” for real-time navigation and decision-making. The Lightweight Architecture Analytics repository could evolve into an “operating system for change.”
- **Rise of Agentic AI:** AI systems with greater autonomy that can perform complex, multi-step tasks—like reasoning, planning, using tools, and learning—with minimal human input. Potential uses in EA include continuous governance checks, proactively detecting and fixing architectural drift, simulating the impact of changes, and optimizing workflows. This also brings the possibility of “self-optimizing organizations,” but introduces new risks regarding control and security.
- **Digital Twins of Organizations (DTOs):** Dynamic, data-rich digital replicas of a company’s operations, processes, and systems. These DTOs would be fueled by real-time data and use AI/GenAI for simulation, prediction, and “what-if” analysis. This aligns perfectly with Grounded Architecture’s goal of having a complete and current understanding of the enterprise. Companies like BMW and UPS are already exploring this.
- **Increased Democratization and Collaboration:** Intuitive AI tools, such as those using natural language interfaces (like chatbots) and automatic data visualizations, can make architectural insights accessible to a wider range of stakeholders. Chatbots that can query EA repositories or AI-generated reports can strengthen Grounded Architecture’s Collaborative Networks.
- **Composable and Modular AI Architectures:** Due to rapid innovation, there will likely be an emphasis on flexible AI system designs that allow easy integration and swapping of components (like different LLMs, vector databases, RAG modules, or AI agents). This fits well with Grounded Architecture’s principle of Adaptability.
- **Evolving Role of the Architect:** The architect’s role may shift more towards higher-level functions: governing AI use, designing ethical safeguards, curating data and AI models, ensuring business alignment, facilitating collaboration, and critically evaluating AI outputs. New roles like “Enterprise AI Architect” may emerge.

- **Vertical AI Specialization:** The trend of AI solutions tailored for specific industries (like healthcare or finance) will likely continue. This will require architects to understand domain-specific AI applications.

11.7: Questions to Consider

- *How can you specifically use GenAI to improve the analytics within your organization? Which data sources are most promising for providing context with RAG?*
- *In what ways could GenAI tools enhance knowledge sharing and communication among your teams and stakeholders? What might be the obstacles to adopting these tools?*
- *Which activities in your team’s current way of working (Operating Model)—like creating ADRs, tracking technical debt, or defining standards—could benefit most from GenAI assistance?*
- *Given the risks of AI making errors (“hallucinations”) or showing bias, what specific human review processes would you need for critical architectural outputs generated by AI?*
- *What are the biggest data privacy and security concerns for using GenAI with your company’s data, and how can you design solutions to effectively reduce these risks?*
- *How can you encourage a culture of responsible AI use and critical thinking about AI outputs among your fellow architects and development teams?*
- *Which specific GenAI use case (e.g., generating diagrams, analyzing requirements, reviewing code) should you try experimenting with first, and how would you measure its success?*
- *How does the Grounded Architecture principle of “Data-Driven Decisions” connect with the need for high-quality data to effectively train and ground GenAI models in your organization?*
- *Looking at future trends like Agentic AI and Digital Twins of Organizations (DTOs), how should you start preparing your skills and your organization’s architecture practice for these advancements?*
- *What ethical guidelines and governance principles are most important for your organization to establish before widely adopting GenAI within your architecture practice?*

11.8: Tables

11.8.1: Grounded Architecture Element vs. GenAI Capability

Grounded Architecture Element	GenAI Capability	Description
Lightweight Architectural Analytics	Data Gathering & Processing	Automates collecting, understanding, and summarizing information from diverse sources like code, cloud costs, support tickets, and documents.
Pattern Recognition & Anomaly Detection		More quickly identifies architectural issues, technical debt, security risks, and old technologies using AI analysis.
Report and Dashboard Generation		Automatically creates reports and dashboards from analyzed data to speed up decision-making.

Grounded Architecture Element	GenAI Capability	Description
Collaborative Networks	Knowledge Management & Sharing	Turns static knowledge (like documents and ADRs) into dynamic, searchable resources using RAG.
	Communication Assistance	Helps draft communications tailored for different audiences, both technical and non-technical.
	Meeting Summarization	Summarizes discussions, decisions, and action items from meeting recordings or notes.
Operating Model	Coding and Documentation Support	Offers AI help for writing code, technical documents, and analyzing requirements.
	Artifact Generation	Helps create architecture diagrams, ADRs, compliance documents, and reports.

Grounded Architecture Element	GenAI Capability	Description
Technical Debt Tracking	Summarizes technical debt issues, flags aging technologies, and helps prioritize fixes based on their impact.	
Due Diligence Acceleration	Speeds up the review of technical documents for mergers, acquisitions, or technology choices.	
Process Standardization	Helps draft standards and governance processes, and can check if proposals meet these standards.	
Strategy Development	Summarizes current situations and suggests future strategies for areas like cloud, data, and technology platforms.	

11.8.2: GenAI Checklist

Best Practice Area	Key Action/Consideration	Relevance to Grounded Architecture
Data Governance	Ensure high-quality, secure, private, managed data. Implement access controls.	Fundamental for reliable Lightweight Analytics & RAG. Supports Data-Driven principle.
Human Oversight	Use Human-in-the-Loop for validation. Architects review AI output. Avoid automation bias.	Reinforces the architect's role in Networks & Operating Model. Upholds Pragmatism.
Model Management	Choose suitable models. Prioritize RAG. Monitor performance, cost. Iterate.	Ensures AI tools effectively support the Operating Model & Analytics.
Security	Design secure applications (e.g., using user identity). Encrypt data. Conduct security reviews.	Protects sensitive data used in Analytics and shared in Collaborative Networks.
Ethics & Fairness	Assess and reduce bias. Ensure transparency and explainability of AI actions.	Ensures fairness in Analytics insights & Operating Model decisions. Builds trust in Collaboration.

Best Practice Area	Key Action/Consideration	Relevance to Grounded Architecture
Governance & Process	Define use cases and objectives. Implement principle-based governance. Document everything.	Structures GenAI within the Operating Model. Aligns AI to goals. Supports Continuous Realignment.
Culture & Literacy	Train users on responsible AI. Foster critical thinking and collaboration.	Enhances how effectively Collaborative Networks use AI. Builds trust.

11.9: To Probe Further: References

11.9.1: Generative AI - General Concepts, Trends & Enterprise Impact

- Generative AI: What Is It, Tools, Models, Applications and Use Cases² - Gartner
- Generative AI Trends For All Facets of Business³ - Forrester
- Generative AI use cases for the enterprise⁴ - IBM Think Blog
- 2024: The State of Generative AI in the Enterprise⁵ - Menlo Ventures
- Beyond ChatGPT: The Future of Generative AI for Enterprises⁶ - Gartner
- GenAI and its impact on the IT infrastructure⁷ - HCLTech Blogs
- Real-world gen AI use cases from the world's leading organizations⁸ - Google Cloud Blog
- The Complete Guide to Generative AI Architecture⁹ - XenonStack Blog

11.9.2: Retrieval-Augmented Generation (RAG)

- What are RAG models? A guide to enterprise AI in 2025¹⁰ - Glean Blog
- 8 Retrieval Augmented Generation (RAG) Architectures You Should Know in 2025¹¹ - Humanloop Blog
- What Is RAG Architecture? A New Approach to LLMs¹² - Cohere Blog

²<https://www.gartner.com/en/topics/generative-ai>

³<https://www.forrester.com/technology/generative-ai/>

⁴<https://www.ibm.com/think/topics/generative-ai-use-cases>

⁵<https://menlovc.com/2024-the-state-of-generative-ai-in-the-enterprise/>

⁶<https://www.gartner.com/en/articles/beyond-chatgpt-the-future-of-generative-ai-for-enterprises>

⁷<https://www.hcltech.com/blogs/genai-and-its-impact-it-infrastructure>

⁸<https://cloud.google.com/transform/101-real-world-generative-ai-use-cases-from-industry-leaders>

⁹<https://www.xenonstack.com/blog/generative-ai-architecture>

¹⁰<https://www.glean.com/blog/rag-models-enterprise-ai>

¹¹<https://humanloop.com/blog/rag-architectures>

¹²<https://cohere.com/blog/rag-architecture>

- What is Retrieval-Augmented Generation (RAG)? A Practical Guide¹³ - K2view
- RAG Is All The Rage — And The Machine Is Getting More Complex¹⁴ - Forrester Blogs
- AI and knowledge management: Why RAG is essential¹⁵ - Outshift - Cisco Blog
- Data Governance for Retrieval-Augmented Generation (RAG)¹⁶ - Enterprise Knowledge
- Enterprise RAG: Bridging Knowledge Gaps with AI-Powered Retrieval¹⁷ - Deepchecks
- Why RAG is a game changer for enterprise knowledge management¹⁸ - HTEC Insights Blog

11.9.3: Agentic AI

- What is Agentic AI? A Practical Guide¹⁹ - K2view
- What Is Agentic Architecture?²⁰ - IBM Think Blog
- What Is Agentic AI?²¹ - NVIDIA Blog
- What is Agentic AI? Definition, Examples and Trends in 2025²² - Aisera Blog
- Tech Navigator: Agentic AI Architecture and Blueprints²³ - Infosys IKI Research
- Agentic AI in enterprise workflow automation²⁴ - IBM Developer

¹³<https://www.k2view.com/what-is-retrieval-augmented-generation>

¹⁴<https://www.forrester.com/blogs/rag-is-all-the-rage-and-the-machine-is-getting-more-complex/>

¹⁵<https://outshift.cisco.com/blog/using-ai-knowledge-management-why-rag-is-essential>

¹⁶<https://enterprise-knowledge.com/data-governance-for-retrieval-augmented-generation-rag/>

¹⁷<https://www.deepchecks.com/bridging-knowledge-gaps-with-rag-ai/>

¹⁸<https://htec.com/insights/blogs/why-rag-is-a-game-changer-for-enterprise-knowledge-management/>

¹⁹<https://www.k2view.com/what-is-agentic-ai/>

²⁰<https://www.ibm.com/think/topics/agentic-architecture>

²¹<https://blogs.nvidia.com/blog/what-is-agentic-ai/>

²²<https://aisera.com/blog/agentic-ai/>

²³<https://www.infosys.com/iki/research/agentic-ai-architecture-blueprints.html>

²⁴<https://developer.ibm.com/articles/agentic-ai-workflow-automation>

- [Agentic AI: The Future of Business Process Automation²⁵](#) - ML Conference Blog
- [From automation to autonomy: Reshaping enterprise architecture with agentic AI and business capability models²⁶](#) - Neudesic Blog
- [Designing Agentic AI Systems, Part 1: Agent Architectures²⁷](#) - Vectorize Blog
- [Agentic AI Architecture: A Deep Dive²⁸](#) - Markovate Blog

11.9.4: Responsible AI, Governance, Best Practices & Security

- [6 Best Practices for Implementing Generative AI²⁹](#) - Iguazio Blog
- [5 Generative AI Best Practices For Enterprise Businesses³⁰](#) - Coveo Blog
- [Responsible AI in a Dynamic Regulatory Environment³¹](#) - Cloud Security Alliance (CSA)
- [Responsible AI in Azure Workloads³²](#) - Microsoft Azure Well-Architected Framework
- [Responsible AI Principles³³](#) - FS-ISAC
- [Responsible AI Architect's Guide³⁴](#) - IndiaAI
- [Best practices to architect secure generative AI applications³⁵](#) - Microsoft Community Hub
- [Building AI Responsibly³⁶](#) - AWS
- [What is Responsible AI³⁷](#) - Azure Machine Learning - Microsoft Learn

²⁵<https://mlconference.ai/blog/agentic-ai-business-process-automation/>

²⁶<https://www.neudesic.com/blog/agentic-ai-business-capability-models/>

²⁷<https://vectorize.io/designing-agentic-ai-systems-part-1-agent-architectures/>

²⁸<https://markovate.com/blog/agentic-ai-architecture/>

²⁹<https://www.iguazio.com/blog/6-best-practices-for-implementing-generative-ai/>

³⁰<https://www.coveo.com/blog/generative-ai-best-practices/>

³¹<https://cloudsecurityalliance.org/artifacts/principles-to-practice-responsible-ai-in-a-dynamic-regulatory-environment>

³²<https://learn.microsoft.com/en-us/azure/well-architected/ai/responsible-ai>

³³https://www.fsisac.com/hubfs/Knowledge/AI/FSISAC_ResponsiveAI-Principles.pdf

³⁴<https://indiaai.gov.in/responsible-ai/pdf/architect-guide.pdf>

³⁵<https://techcommunity.microsoft.com/blog/microsoft-security-blog/best-practices-to-architect-secure-generative-ai-applications/4116661>

³⁶<https://aws.amazon.com/ai/responsible-ai/>

³⁷<https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai?view=azureml-api-2>

- Responsible AI by design: Building a framework of trust³⁸ - The Enterprisers Project
- EU AI Act: A Complete Guide for Enterprise Architects³⁹ - Ardoq Knowledge Hub
- 10 Best Practices for Scaling Generative AI⁴⁰ - SnapLogic Community
- Responsible AI Guidelines⁴¹ - Defense Innovation Unit (DIU)

11.9.5: GenAI & Enterprise Architecture / Solution Architecture

- Real-Time Enterprise Architecture In The Age Of AI⁴² - Forrester Blogs
- AI And GenAI Are Game-Changers For Enterprise Architecture Leaders⁴³ - Forrester Blogs
- The Future of Enterprise Architecture in an AI-Driven World⁴⁴ - Techstrong.ai
- GenAI with IT Architecture: Building intelligent foundations together⁴⁵ - Cognizant Insights Blog
- Architects: Jump In To Generative AI⁴⁶ - Forrester Blogs
- What is AI in solutions architecture?⁴⁷ - Arphie Glossary
- Generative AI and EA: Modeling the Enterprise⁴⁸ - Ardoq Blog

³⁸<https://enterprisersproject.com/article/2022/12/responsible-ai-design-building-framework-trust>

³⁹<https://www.ardoq.com/knowledge-hub/eu-ai-act>

⁴⁰<https://community.snaplogic.com/t5/ai-ml-genai-app-builder/gartner-10-best-practices-for-scaling-generative-ai/m-p/25488>

⁴¹<https://www.diu.mil/responsible-ai-guidelines>

⁴²<https://www.forrester.com/blogs/the-augmented-architect-real-time-enterprise-architecture-in-the-age-of-ai/>

⁴³<https://www.forrester.com/blogs/ai-and-genai-are-game-changers-for-enterprise-architecture-leaders/>

⁴⁴<https://techstrong.ai/articles/the-future-of-enterprise-architecture-in-an-ai-driven-world/>

⁴⁵<https://www.cognizant.com/ch/de/insights/blog/articles/genai-with-it-architecture>

⁴⁶<https://www.forrester.com/blogs/architects-jump-in-to-generative-ai/>

⁴⁷<https://www.arphie.ai/glossary/ai-in-solutions-architecture>

⁴⁸<https://www.ardoq.com/blog/ai-enterprise-architecture-modeling>

- The Role of AI in Enterprise Architecture: A Future Outlook⁴⁹ - ValueBlue Blog
- A Pragmatic Perspective on GenAI in Solution Architecture⁵⁰ - EPAM Insights Blog
- The future of enterprise architecture and AI integration⁵¹ - Bizzdesign Blog
- Digital Experience - Gen AI powered Enterprise Architecture⁵² - Infosys Blogs
- AI Architecture Design - Azure⁵³ - Microsoft Learn
- Enterprise Architecture and Digital Transformation Trends for 2025⁵⁴ - Orbus Software Blog
- How Generative AI is Revolutionizing Enterprise Architecture – Must-Read Insights!⁵⁵ - Reddit (r/EnterpriseArchitect)
- Has anyone used AI Tools for Solution Design and Architecture (I will not promote)⁵⁶ - Reddit (r/startups)
- 9 AI Architecture Design Challenges & Solutions⁵⁷ - RTS Labs
- Unlocking Effectiveness: How Generative AI is Revolutionizing Enterprise Architecture⁵⁸ - Protiviti TC Blog
- Generative AI and Enterprise Architecture: Roadmapping the Future⁵⁹ - Ardoq Blog
- Game-changing Enterprise Architecture Generative AI features you can't miss!⁶⁰ - Bizzdesign Blog

⁴⁹<https://www.valueblue.com/blog/the-role-of-ai-in-enterprise-architecture-a-future-outlook>

⁵⁰<https://www.epam.com/insights/blogs/a-pragmatic-perspective-on-generative-ai-in-solution-architecture>

⁵¹<https://bizzdesign.com/blog/the-future-of-enterprise-architecture-and-ai-integration/>

⁵²<https://blogs.infosys.com/digital-experience/emerging-technologies/gen-ai-powered-enterprise-architecture.html>

⁵³<https://learn.microsoft.com/en-us/azure/architecture/ai-ml/>

⁵⁴<https://www.orbussoftware.com/resources/blog/detail/enterprise-architecture-and-digital-transformation-trends-for-2025>

⁵⁵https://www.reddit.com/r/EnterpriseArchitect/comments/1d15i8l/how_generative_ai_is_revolutionizing_enterprise/

⁵⁶https://www.reddit.com/r/startups/comments/1iwy7ns/has_anyone_used_ai_tools_for_solution_design_and/

⁵⁷<https://rtslabs.com/ai-solution-architecture-design>

⁵⁸<https://tcblog.protiviti.com/2024/06/18/unlocking-effectiveness-how-generative-ai-is-revolutionizing-enterprise-architecture/>

⁵⁹<https://www.ardoq.com/blog/ai-enterprise-architecture-roadmapping>

⁶⁰<https://bizzdesign.com/blog/bizzdesign-horizon-generative-ai-features/>

- The Future of Generative AI: Empowering Enterprise Architects⁶¹
- Vultr Blogs
- Navigating AI Implementation: The Case for an Enterprise AI Architect⁶² - BCG Platinion Insights

11.9.6: GenAI Use Cases & Tools (Specific Areas)

Cloud Operations:

- The Role of Generative AI in Enhancing Cloud Operations: Real Use Cases⁶³ - RTInsights

Code Review & Development:

- Generative AI Use Cases and Resources⁶⁴ - AWS
- AI Code Review: An Engineering Leader's Survival Guide⁶⁵ - LinearB Blog
- AI Code Reviews⁶⁶ - GitHub Resources
- AI-Driven Code Review: Enhancing Developer Productivity and Code Quality⁶⁷ - CACM Blog
- AI Code Reviews⁶⁸ - CodeRabbit
- AI Code Review⁶⁹ - IBM Think Insights
- Code Reviews with AI a developer guide⁷⁰ - foojay.io

Documentation & Diagramming:

⁶¹<https://blogs.vultr.com/the-future-of-generative-ai-empowering-enterprise-architects>

⁶²<https://www.bcgplatinion.com/insights/enterprise-ai-architect>

⁶³<https://www.rtinsights.com/the-role-of-generative-ai-in-enhancing-cloud-operations-real-use-cases/>

⁶⁴<https://aws.amazon.com/ai/generative-ai/use-cases/>

⁶⁵<https://linearb.io/blog/ai-code-review>

⁶⁶<https://github.com/resources/articles/ai/ai-code-reviews>

⁶⁷<https://cacm.acm.org/blogcacm/ai-driven-code-review-enhancing-developer-productivity-and-code-quality/>

⁶⁸<https://www.coderrabbit.ai/>

⁶⁹<https://www.ibm.com/think/insights/ai-code-review>

⁷⁰<https://foojay.io/today/code-reviews-with-ai-a-developer-guide/>

- AI Architecture Diagram Generator⁷¹ - Eraser.io
- SWAPP integrates advanced AI with human expertise to automate documentation and modeling tasks⁷² - Swapp
- Klarify Architect - Transform Your Documentation and Processes⁷³ - Klarify.ai
- 20 AI Tools for Architects⁷⁴ - Part3 Blog
- Now Assist for Enterprise Architecture (EA)⁷⁵ - ServiceNow Store
- Generative AI for Architecture Firms⁷⁶ - Invoke
- AI Tool that can generate Architecture Diagrams? : r/AWSCertifications⁷⁷ - Reddit (r/AWSCertifications)
- Generative AI on Architecture Diagram Creation : Part-2⁷⁸ - RandomTrees Blog
- AI tools you use as an architect? : r/softwarearchitecture⁷⁹ - Reddit (r/softwarearchitecture)
- AI Documentation Generator⁸⁰ - Bito AI Blog
- 15 Top AI Tools for Architects and Designers⁸¹ - Architizer Journal
- Top 16 AI Tools for Architects in 2025⁸² - Enscape Blog

Requirements Analysis:

- “Revolutionizing Systems Engineering: The Role of AI in Requirements Analysis”⁸³ - DEV Community
- 4 Ways AI Is Transforming Requirements Management⁸⁴ - Altium 365 Resources

⁷¹<https://www.eraser.io/ai/architecture-diagram-generator>

⁷²<https://swapp.ai/>

⁷³<https://www.klarify.ai/architect>

⁷⁴<https://www.part3.io/blog/best-ai-tools-architects>

⁷⁵<https://tpp.servicenow.com/store/app/4afc6f621b646a50a85b16db234bcbe0>

⁷⁶<https://www.invoke.com/solutions/generative-ai-architecture>

⁷⁷https://www.reddit.com/r/AWSCertifications/comments/1drdhij/ai_tool_that_can_generate_architecture_diagrams/

⁷⁸<https://randomtrees.com/blog/generative-ai-on-architecture-diagram-creation-part-2/>

⁷⁹https://www.reddit.com/r/softwarearchitecture/comments/1dk5dn5/ai_tools_you_use_as_an_architect/

⁸⁰<https://bito.ai/blog/ai-documentation-generator/>

⁸¹<https://architizer.com/blog/practice/tools/top-ai-tools-for-architects-and-designers/>

⁸²<https://blog.enscape3d.com/ai-tools-for-architects>

⁸³https://dev.to/gilles_hamelink_ea9ff7d93/revolutionizing-systems-engineering-the-role-of-ai-in-requirements-analysis-29ja

⁸⁴<https://resources.altium365.com/p/ai-transforming-requirements-management>

- [Using AI for requirements analysis: A case study⁸⁵](#) - Thoughtworks Insights Blog

Technical Debt Management:

- [9 Tools to Measure Technical Debt in 2025⁸⁶](#) - CodeAnt AI Blog
- [Top 10 Tools to Manage and Track Technical Debt in 2025⁸⁷](#) - ClickUp Blog
- [Artificial Intelligence for Technical Debt Management in Software Development⁸⁸](#) - arXiv
- [Tackling Technical Debt with Generative AI⁸⁹](#) - Encora Insights
- [Managing Technical Debt with AI-Powered Productivity Tools: A Complete Guide⁹⁰](#) - Qodo Blog
- [Reducing Technical Debt with AI⁹¹](#) - Concord USA Blog

⁸⁵<https://www.thoughtworks.com/en-us/insights/blog/generative-ai/using-ai-requirements-analysis-case-study>

⁸⁶<https://www.codeant.ai/blogs/tools-measure-technical-debt>

⁸⁷<https://clickup.com/blog/technical-debt-tools/>

⁸⁸<https://arxiv.org/pdf/2306.10194>

⁸⁹<https://insights.encora.com/insights/tackling-technical-debt-with-generative-ai>

⁹⁰<https://www.qodo.ai/blog/managing-technical-debt-ai-powered-productivity-tools-guide/>

⁹¹<https://www.concordusa.com/blog/reducing-technical-debt-with-ai>