



Grounded Architecture

Part I: 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: Foundations	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
Part II: Grounded Architecture Framework: Operating Model	94
7: Operating Model: Introduction	95
8: Operating Model: General Principles	99
8.1: Examples of Architecture Activities	102
8.2: Guiding Principles for Architectural Excellence: Policies, Autonomy, and Engagement	105
8.3: Embracing Diversity	112
8.4: Setting Boundaries	114
8.5: Final Thoughts	117
8.6: To Probe Further	118
8.7: Questions to Consider	119
9: Governance: Nudge, Taxation, Mandates	120
9.1: Nudging as a Governance Tool	124
9.2: Taxation (Economic Incentives) as a Governance Mechanism	127

9.3: Mandates and Bans as Governance Tools	130
9.4: Questions to Consider	133
10: Putting Everything Together: Transforming Organizations with Grounded Architecture	134
10.1: Executing at Scale	137
10.2: Maximizing Organizational Learning	145
11: Bonus: Leveraging Generative AI within Grounded Architecture Framework	148
11.1: What Generative AI Can Do for Today's Architect	151
11.2: Integrating GenAI with Grounded Architecture	155
11.3: Practical Examples: GenAI in Action	158
11.4: Balancing Benefits, Risks, and Challenges	161
11.5: Making GenAI Work Within Grounded Architecture	164
11.6: The Evolving Landscape	166
11.7: Final Thoughts	170
11.8: To Probe Further	172
11.9: Questions to Consider	173
11.10: Tables	174

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.

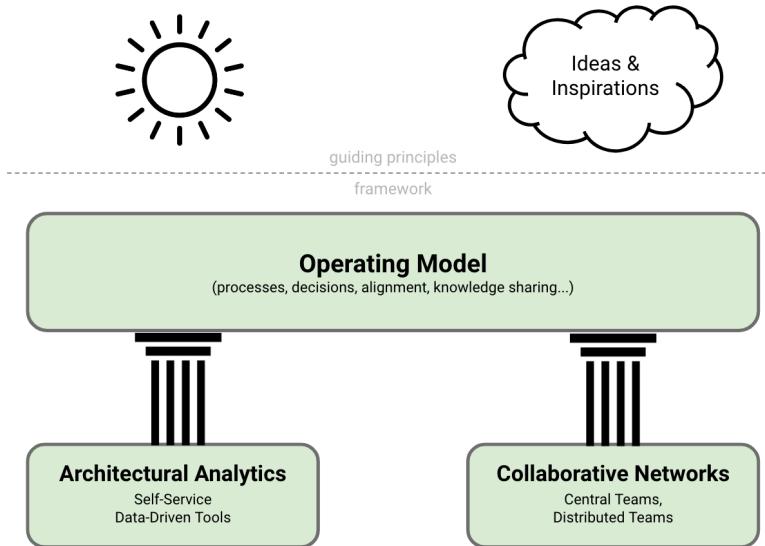


Figure 1: Grounded Architecture Overview

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.

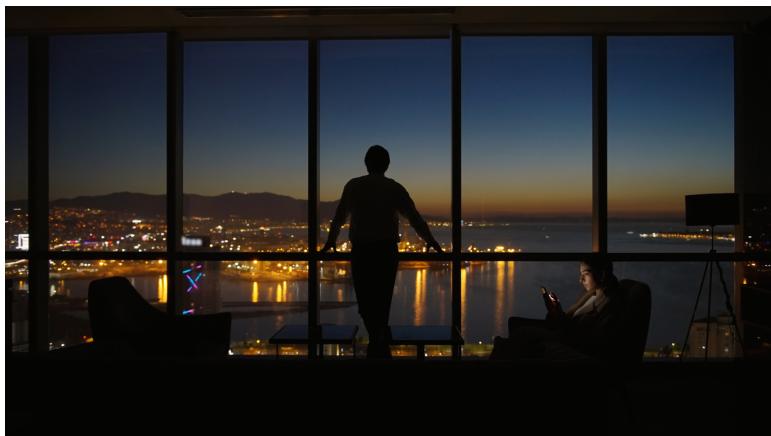


image by brightstars from istock

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.

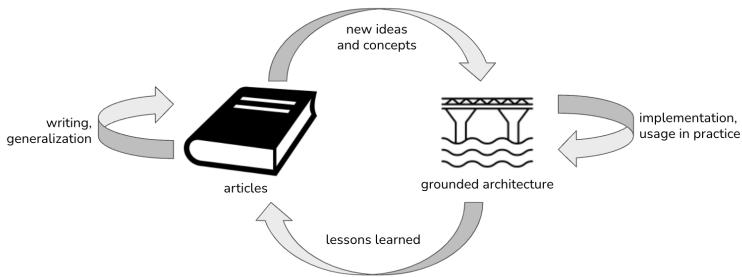


Figure 2: Writing helped me refine my thinking, evolve tools and principles, and share lessons learned.

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**.

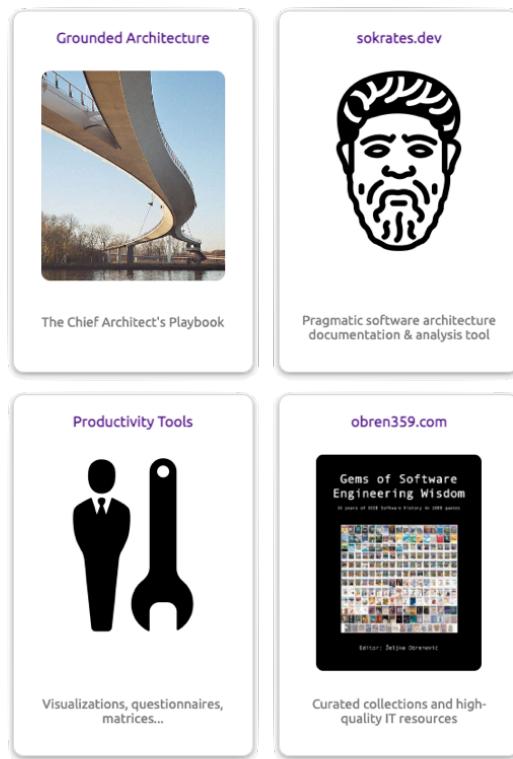


Figure 3: *Grounded Architecture* is part of a collection of open-source tools and learning resources developed over the past 10 years.

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: Foundations

4: Grounded Architecture Framework: Foundations

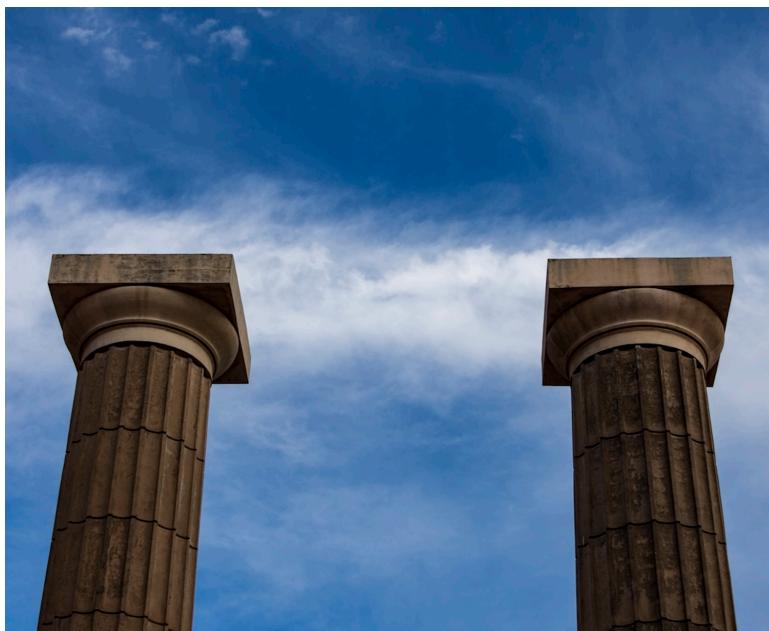


image by dario truco from istock

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>

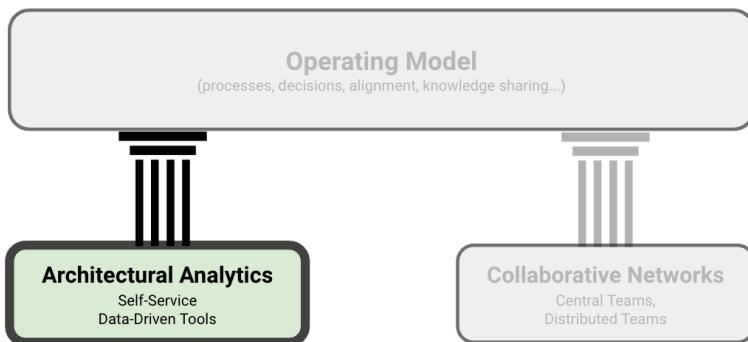


Figure 1: Grounded Architecture Framework – Lightweight Architectural Analytics

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.

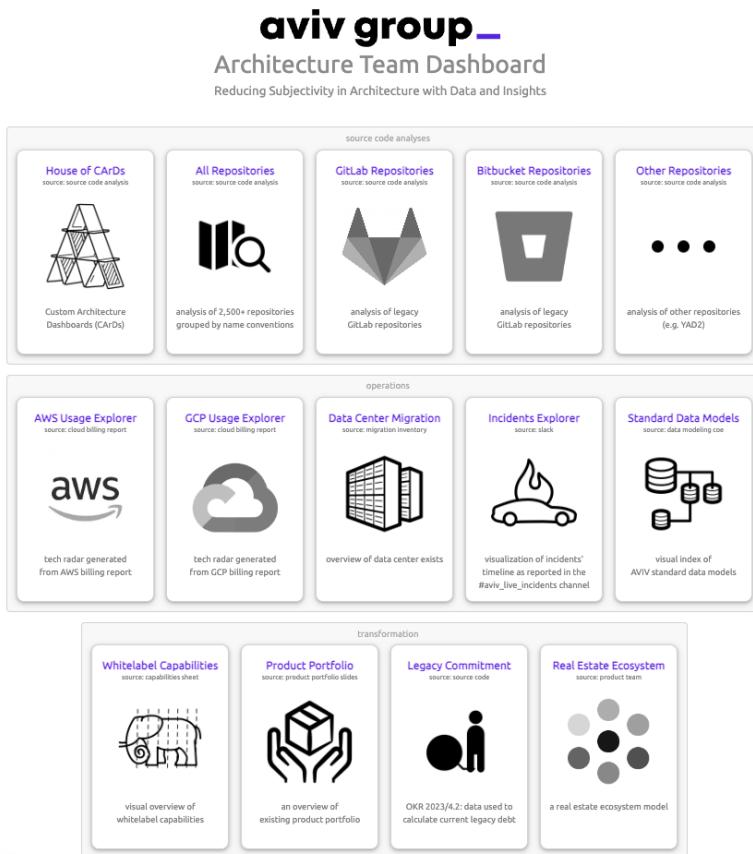


Figure 2: Start page of the architecture analytics dashboard used at AVIV Group.

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:

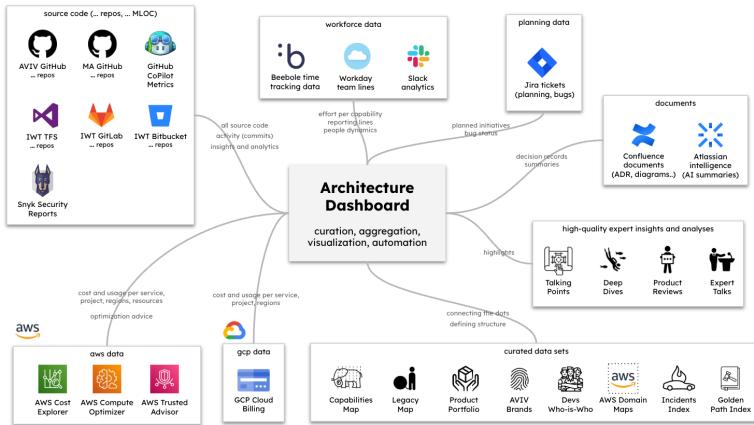


Figure 3: Overview of core data sources powering the architecture dashboard.

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>

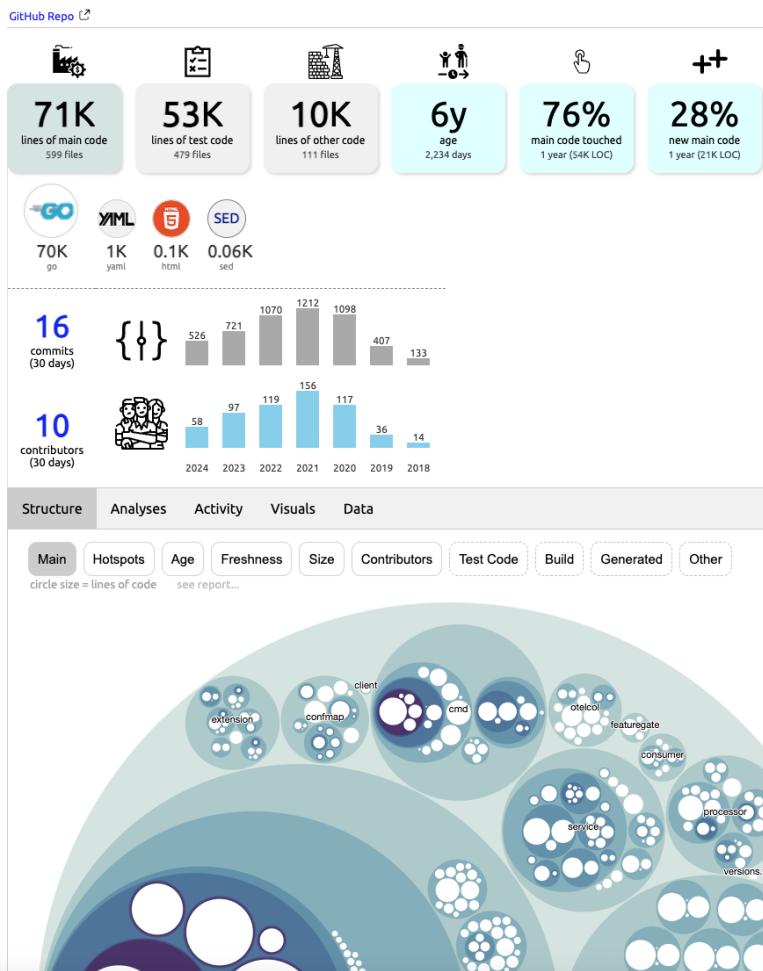


Figure 4: Screenshot from a Sokrates report dashboard.

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/>

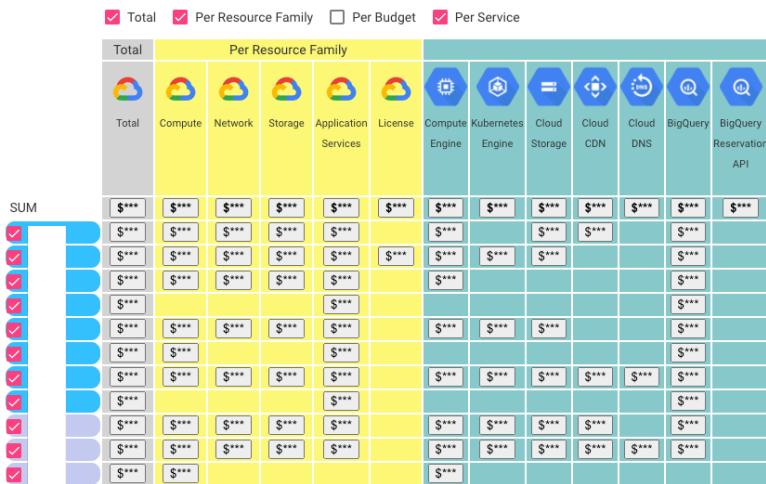


Figure 5: An example of a cloud usage explorer.

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**.

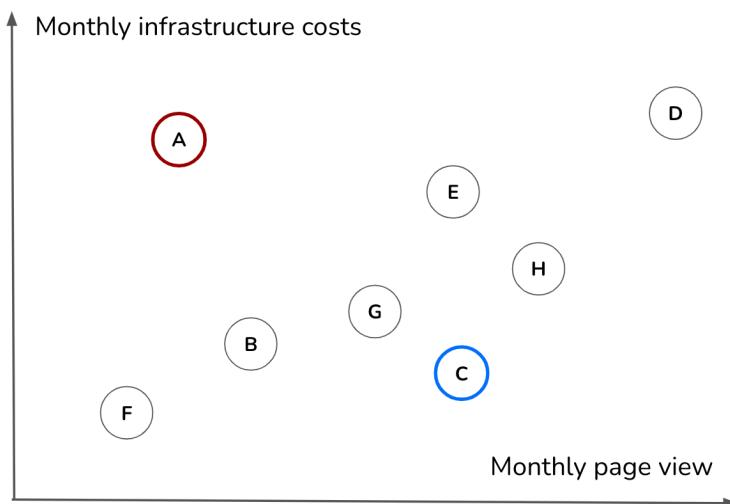


Figure 6: Combining data from different sources (e.g., cloud billing reports and vibrancy or revenue can lead to new insights (e.g., identifying inefficiencies in the application portfolio).

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.

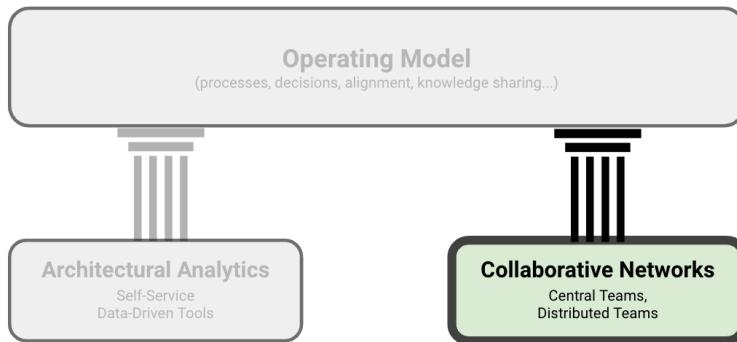


Figure 1: *The Grounded Architecture framework – Collaborative Networks.*

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]).

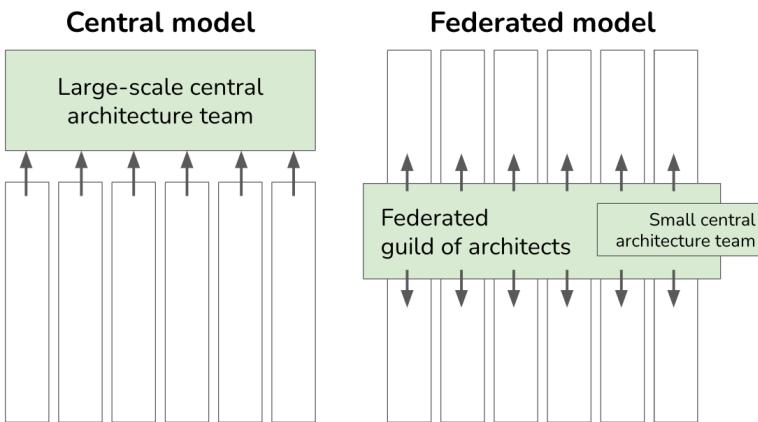


Figure 2: Central vs. Federated Architecture Practice.

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?*

Part II: Grounded Architecture Framework: Operating Model

7: Operating Model: Introduction

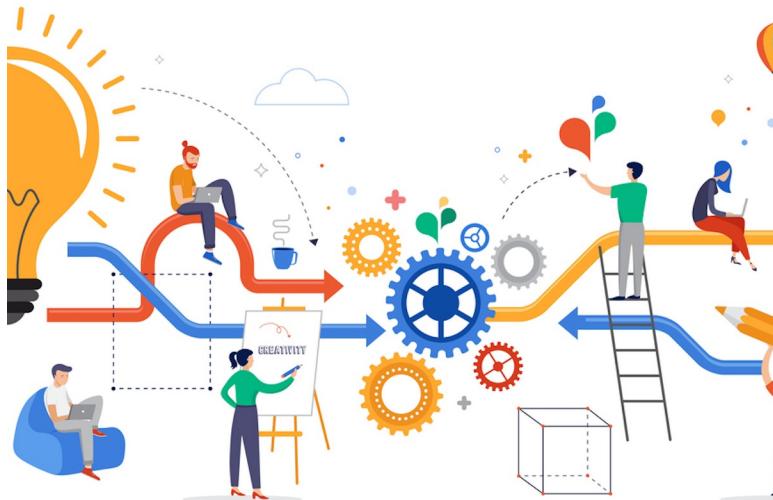


image by ma_rish from istock

IN THIS SECTION, YOU WILL: Get an overview of the Operating Model in the Grounded Architecture framework that provides a flexible, principle-driven structure for aligning architecture with business goals.

KEY POINTS:

- The Operating Model is a core part of the Grounded Architecture framework, guiding how architecture functions within an organization through principles, governance, and practices.
- It is designed to evolve with the organization while ensuring architectural activities remain aligned with business objectives.
- We introduce four main areas of the operating model: General Principles, Governance (Nudge, Taxation, Mandates), Transformation Support (in support of Grounded Architecture Goals), and Generative AI.
- The model's success depends on robust data (Lightweight Architectural Analytics) and strong collaboration (Collaborative Networks), not just processes and structure.

In this section, we will explore the **Operating Model**, a crucial element of the Grounded Architecture framework (see Figure 1). The Operating Model offers a **structured approach** to how an architecture practice operates within an organization. It outlines the **principles, governance structures, and operational practices** that guide architects in their daily tasks.

Designed to be **flexible and adaptive**, the Operating Model evolves alongside the organization while consistently focusing on **delivering value** through architecture. It ensures that architectural efforts align with business goals and are effectively coordinated across teams and domains.

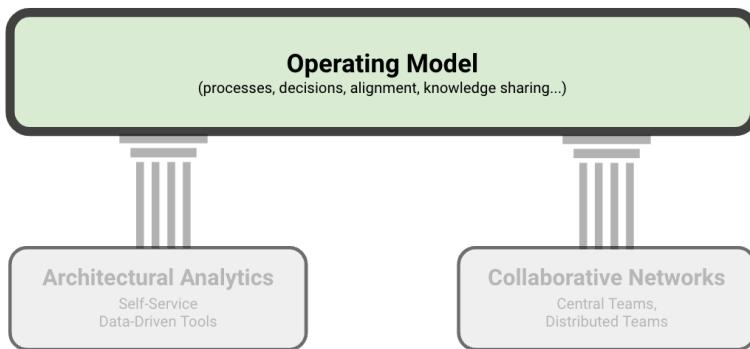


Figure 1: *The Grounded Architecture framework – Operating Model.*

The Operating Model comprises four main areas:

- **General Principles**

A set of guiding values and actionable principles to help architects operate effectively. These principles highlight the importance of utilizing data and networks to create meaningful, measurable impact across the organization.

- **Nudge, Taxation, Mandates – Governance**

A balanced approach to IT governance, blending three styles:

- **Nudging:** Subtle guidance through defaults and visibility
- **Taxation:** Economic incentives to influence behavior
- **Mandates:** Clear rules and standards to ensure compliance

This flexible model allows organizations to strike a balance between **innovation and consistency** as well as **autonomy and alignment**.

- **Transforming with Grounded Architecture¹** (in support of Grounded Architecture Goals) An examination of how Grounded Architecture facilitates meaningful organizational transformation, which includes:

- Executing at scale
- Enhancing data-driven decision-making
- Maximizing alignment and learning
- Increasing adaptability

- **Bonus: Leveraging Generative AI in Architecture²**

¹transforming

²gen-ai

Practical methods for utilizing Generative AI as a tool for enhancement—automating routine tasks, improving collaboration, and accelerating analysis—while acknowledging the necessity of **responsible use, human oversight, and strong governance.**

While the Operating Model offers a wealth of structures and ideas, it is essential to emphasize one key point: **no operating model functions in isolation.**

Without **healthy Lightweight Architectural Analytics** (data) and **strong Collaborative Networks** (people), any operating model risks becoming an **ivory tower**—a set of well-meaning policies and opinions that are disconnected from the realities of daily work.

In summary: **people and data come first—process comes second.**

8: Operating Model: General Principles



image by maria castellanos from istock

IN THIS SECTION, YOU WILL: Understand that an effective architecture practice must be tailored to organizational needs, grounded in collaboration and clear rules of engagement, and focused on enabling teams through strategic alignment, distributed decision-making, and supported standards like golden paths.

KEY POINTS:

- No one-size-fits-all architecture practice: Organizations must customize their architecture practices to address specific challenges and contexts, focusing on resolving the organization's most pressing issues.
- Successful architectural work depends on collaboration with teams, distributed decision-making, and clearly defined "rules of engagement." Architects should empower teams, get involved early, avoid bureaucratic delays, and facilitate alignment with strategic goals.
- "Golden Paths"—clear, opinionated, and supported solutions—are essential for reducing fragmentation, increasing consistency, and guiding teams toward efficient and aligned development practices.
- Important architecture activities include designing decision frameworks, supporting projects and legacy modernization, tracking technical debt, conducting platform analyses, and defining strategic technology directions—all while being integrated into team workflows.
- Architectural support is most effective when teams meet baseline criteria (e.g., stakeholder awareness, skills, documentation literacy). Establishing expectations and defining boundaries ensures that architectural efforts are scalable and focused on areas with the most significant impact.

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

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

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.

This section outlines some lessons I learned when defining IT architecture operating models. The Operating Model 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.

8.1: 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.

8.2: 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.

8.2.1: 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 1).

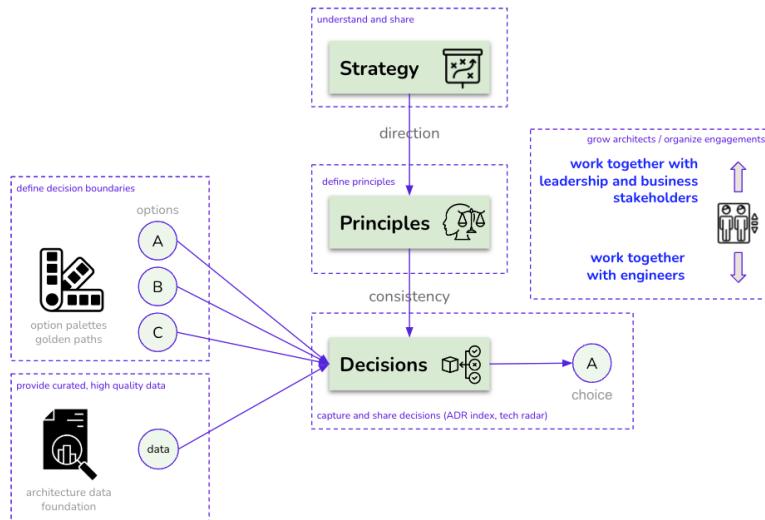


Figure 1: A common operating framework I typically use for Grounded Architecture activities.

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.

8.2.2: 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 2) 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 2: 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).

8.2.3: 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.

8.2.4: 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.

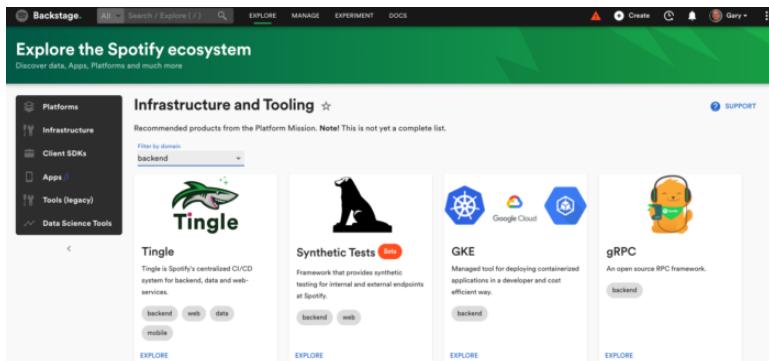


image by engineering.atspotify.com

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.

8.3: 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.

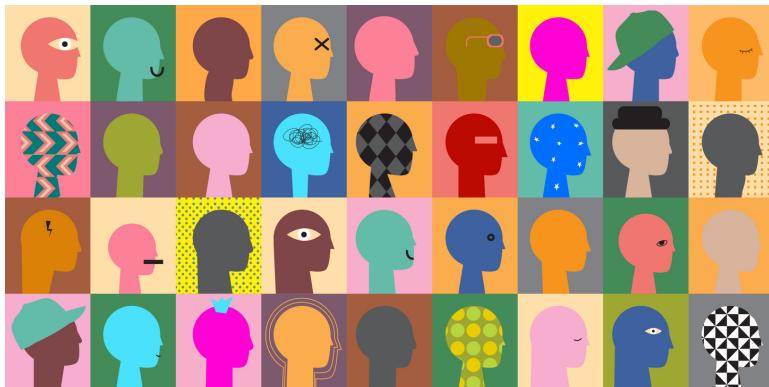


image by annaspoka from istock

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.

8.4: 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!

8.5: Final Thoughts

Here are **five key points** to consider when defining an architecture practice and its operating model within your organization:

1. **No One-Size-Fits-All Architecture Practice:** Organizations must customize their architecture practices to address specific challenges and contexts. It's ineffective to place architects in disarray and expect positive results—architecture should focus on resolving the organization's most pressing issues.
2. **Collaborative and Distributed Operating Models:** Successful architectural work depends on collaboration with teams, distributed decision-making, and clearly defined “rules of engagement.” Architects should empower teams, get involved early, avoid bureaucratic delays, and ensure alignment with strategic goals.
3. **Golden Paths and Standardization Promote Alignment:** “Golden Paths”—clear, opinionated, and supported solutions—are essential for reducing fragmentation, increasing consistency, and guiding teams toward efficient and aligned development practices.
4. **Architecture Activities Must Be Grounded and Strategic:** Important architecture activities include designing decision frameworks, supporting projects and legacy modernization, tracking technical debt, conducting platform analyses, and defining strategic technology directions—all while being integrated into team workflows.
5. **Set Clear Expectations and Boundaries for Effective Support:** Architectural support is most effective when teams meet baseline criteria (e.g., stakeholder awareness, skills, documentation literacy). Establishing expectations and defining boundaries ensures that architectural efforts are scalable and focused on areas with the most significant impact.

8.6: 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/>

8.7: 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?*

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: Putting Everything Together: 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 effectively implemented, it can transform an organization's operations, especially in complex and fast-paced environments. The positive impact of this framework aligns closely with our defined goals, which include:

- Enabling the execution of architecture practices at scale
- Increasing the adaptability of architecture practices
- Enhancing decision-making quality with data
- Maximizing organizational alignment
- Enhancing organizational learning

10.1: Executing at Scale

Our initial challenge was to provide architectural support for **hundreds of teams and thousands of projects**, all functioning in a dynamic environment filled with significant complexity and diversity. Traditional methods simply do not scale effectively in such settings. The Grounded Architecture framework addresses this challenge by combining lightweight data tools, collaborative practices, and a modern operating model.

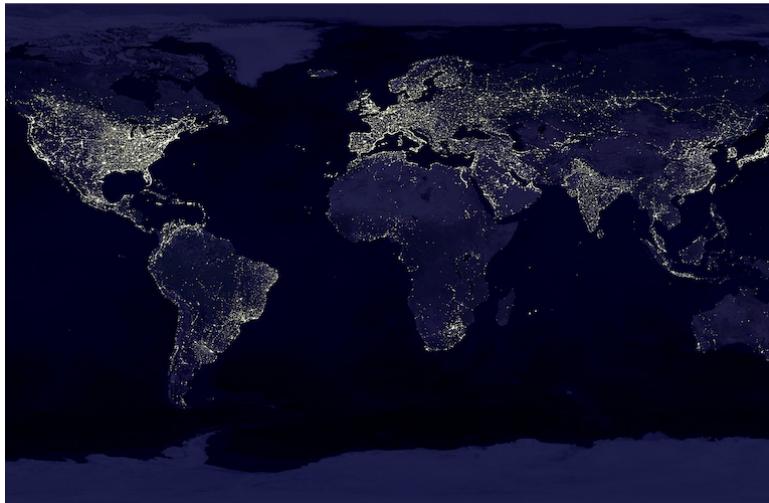


image by wikiimages from pixabay

10.1.1: Lightweight Architectural Analytics

Lightweight Architectural Analytics is a crucial enabler for scalability. It supports large-scale architectural practices through:

- **Self-Service Data Access**

By providing architecture-related insights and tools via internal

portals, teams can access the data they need without waiting for an architect. This approach removes bottlenecks and reduces overhead.

- **Elimination of Routine Meetings**

Traditionally, architects spend a significant amount of time sharing knowledge about system dependencies and the overall architecture through meetings and messages. With analytics dashboards in place, much of this can be automated or accessed as needed—freeing up valuable time and reducing communication overhead. Each new data application deployed saves hours of meeting time and message traffic.

- **Automation of Data Processes**

Manual processes do not scale effectively. By automating the collection, analysis, and distribution of architecture-related data, we ensure consistency, accuracy, and scalability—allowing insights to flow without manual intervention.

10.1.2: Collaborative Networks

The **Collaborative Networks** component enhances execution at scale by strengthening the social structure around architecture:

- **Building Connections Across the Organization**

Relationships are important. Collaborative networks create and maintain channels between teams, departments, and decision-makers, accelerating alignment and mutual understanding.

- **Speeding Up Alignment**

When trust and familiarity are built into the system, teams can align more quickly, resolve conflicts more effectively, and move forward without unnecessary delays.

- **Facilitating Shared Decisions**

Instead of relying solely on top-down directives, collaborative networks enable teams to **co-create solutions**. This distributed decision-making model fosters ownership, increases speed, and enhances engagement.

10.1.3: Operating Model

The **Operating Model** brings everything together into a coherent way of working:

- **Distributed Decision-Making**

By allowing teams to make decisions within their own context, the organization avoids bottlenecks and becomes more agile. This also ensures that decisions are made closer to where the actual work takes place, leading to better and faster outcomes.

- **Collaborative Governance**

A modern operating model emphasizes shared responsibility and cross-functional collaboration. This approach ensures that multiple projects and initiatives can be executed in parallel without overwhelming central teams or processes.

The Grounded Architecture framework includes several design choices that make it inherently flexible and easy to evolve. These core components—analytics, networks, and an operating model—enable the architecture practice to pivot quickly in response to new challenges or opportunities.

10.1.4: Lightweight Architectural Analytics

Lightweight Architectural Analytics plays a pivotal role in supporting adaptability:

- **Automation and Extensibility** We built our tools using lean, modular components that could be easily extended or reconfigured. For example, after mergers or acquisitions, we were able to integrate hundreds of new code repositories and data sources in hours or days rather than weeks. Automation ensures that data remains current and provides real-time insights that evolve with the organization.
- **Dynamic Views for Changing Structures** Our analytics were designed to adjust quickly to shifts in team structures, naming conventions, or project groupings. For instance, we created on-the-fly aggregation tools that allowed us to group data using metadata like repository tags, ensuring that reporting always matched the current organizational structure.

10.1.5: Collaborative Networks

Collaborative Networks enhance adaptability through decentralized strength and shared capacity:

- **Capacity Redistribution** The central team could step in when local teams faced spikes in demand or lacked architectural expertise. This **redistribution of knowledge and support** ensured that architectural guidance remained uninterrupted even during high-stress periods.
- **Decentralized Support Model** A connected and empowered network of architects allowed us to **deliver architecture services locally**—without relying on a centralized team. This **distributed resilience** meant that the organization could scale and shift without weakening architectural oversight.

10.1.6: Operating Model

The **Operating Model** reinforces adaptability by making decision-making more fluid and resilient:

- **Flexible Decision-Making Structures** Rather than a rigid top-down process, decision-making was distributed across the organization. This distribution meant teams could make architectural decisions **where and when needed**, reducing bottlenecks and accelerating delivery.
- **Strategic Delegation** Senior architects could focus on high-impact areas—such as cloud strategy, platform architecture, or major business transitions—because routine decisions were safely delegated to teams. This delegation was made possible by grounding the architectural practice in data transparency and strong human networks, ensuring alignment even in a decentralized model.

To transition from **opinion-driven debates** to **data-informed decisions**, we emphasize the interplay of three foundational components: **Lightweight Architectural Analytics**, **Collaborative Networks**, and a modern **Operating Model**. Together, these elements create an environment where data is accessible, shared, and actionable.

10.1.7: Lightweight Architectural Analytics

Lightweight Architectural Analytics ensures that high-quality, relevant data informs architectural choices. It promotes data-driven decision-making through:

- **Systematic Data Collection:** Gathering key metrics from internal systems, repositories, platforms, and external benchmarks.
- **Structured Data Management:** Organizing and maintaining easily accessible data repositories, enabling architects to locate and utilize necessary information swiftly.
- **Meaningful Data Analytics:** Applying analytical tools to uncover trends, bottlenecks, or anomalies that guide design decisions and trade-offs.
- **Real-Time Data Availability:** Continuously updating dashboards and tools so decision-makers always work with **current, relevant insights**.

These analytics ensure that discussions are based on facts, not assumptions, driving **clearer, faster, and more defensible decisions**.

10.1.8: Collaborative Networks

Access to data is one thing; ensuring that the right people can use it effectively is another. **Collaborative Networks** ensure a smooth flow of data to where decisions are being made:

- **Expert Communities:** Connecting architects, engineers, and stakeholders across teams to integrate diverse perspectives and specialized knowledge into discussions.
- **Collaboration Platforms:** Utilizing tools that support sharing dashboards, annotating findings, and co-creating decisions in real-time.
- **Data Literacy and Training:** Providing training and support so decision-makers can confidently and accurately interpret and apply data.
- **Stakeholder Engagement:** Encouraging regular interaction with decision-makers to ensure that analytics are not just available but also **trusted and utilized**.

This networked approach guarantees that insights are **not siloed**, but embedded into the decision-making culture throughout the organization.

10.1.9: Operating Model

A modern **Operating Model** provides the governance and processes necessary to make data-driven decision-making **repeatable and scalable**:

- **Decision-Making Processes:** Formalizing how architectural proposals are evaluated using data—moving beyond buzzwords to clear, comparative assessments.
- **Integrated Data Workflows:** Incorporating data sources and analytics directly into architecture workflows, ensuring that data is a starting point rather than an afterthought.
- **Economic Risk Modeling:** Connecting technical decisions to business impact by quantifying trade-offs, risks, and costs, helping to ensure architecture supports **strategic outcomes**.
- **Decision Support Tools:** Utilizing visual models, dashboards, and scenario tools to guide discussions and clearly illustrate the implications of each option.

Misalignment is common in large, complex organizations. It often arises from siloed teams, diverse objectives, geographical separation, and inconsistent communication. Grounded Architecture addresses these challenges through three core elements that enhance transparency, collaboration, and shared understanding.

10.1.10: Lightweight Architectural Analytics

Lightweight Architectural Analytics improves alignment by creating transparency:

- By making architecture-related data **readily accessible and understandable**, teams across different departments can operate from a shared foundation.
- Real-time visibility into systems, dependencies, and technical trends allows teams to align their decisions with organizational goals and each other.

- When everyone can access the same data, misunderstandings decrease and trust increases—both essential for alignment.

Transparency not only informs but also unifies. It helps anchor conversations in facts, creating a shared frame of reference for decisions and priorities.

10.1.11: Collaborative Networks

Collaborative Networks enhance alignment by **building connections across the organization**:

- These networks enable cross-functional collaboration, allowing architects, engineers, and product leaders to work together toward shared goals.
- Collaborative networks create a **common organizational rhythm** by connecting individuals and teams across locations, domains, and hierarchies.
- Regular interaction and shared experiences promote alignment in actions, culture, values, and intent.

A connected organization is an aligned organization where knowledge, practices, and goals are continuously harmonized.

10.1.12: Operating Model

The **Operating Model** improves alignment by embedding it into both **decision-making and communication** processes:

- **Pre-Decision Alignment:** The operating model encourages early engagement across teams before significant decisions are made. This alignment minimizes duplicated efforts and surfaces dependencies, and ensures that initiatives are complementary rather than conflicting.
- **Post-Decision Dissemination:** After decisions are made, the operating model supports broad and consistent sharing. Insights, outcomes, and strategic choices are disseminated throughout the organization using Collaborative Networks, ensuring that everyone is informed, aligned, and able to act.

This two-part approach ensures that alignment is maintained **before and after decisions**, making architecture a continuous force for cohesion.

Through transparent analytics, collaborative structures, and a strategic operating model, Grounded Architecture helps unify an organization. It ensures that decisions are not made in isolation, teams do not drift apart, and everyone moves together toward a common goal.

By minimizing misalignments, architecture becomes more than just a technical discipline—it becomes a “strategic enabler of organizational unity.”

10.2: Maximizing Organizational Learning

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

The fifth and final goal of the Grounded Architecture framework is to ensure that architecture serves as a **driver of continuous learning** throughout the organization. In a rapidly changing technology landscape, **learning is not optional; it’s essential**. Architects play a key role by identifying emerging technologies, analyzing trends, and recommending practical upgrades that align with business needs.



image by rawpixel from istock

The Grounded Architecture framework fosters organizational learning through three core pillars: Lightweight Architectural Analytics, Collaborative Networks, and a robust Operating Model. Together, these components ensure that **learning is continuous, actionable, and embedded** in daily practices.

10.2.1: Lightweight Architectural Analytics

Lightweight Architectural Analytics accelerates learning by enabling exploration, experimentation, and insight generation:

- **Facilitating Exploration and Reflection**

These tools provide teams with meaningful, contextual data that helps them explore new technologies safely. By allowing quick experiments—such as proofs-of-concept with new cloud services—and analyzing real usage data (like cost or performance), teams gain insightful feedback that guides their decisions.

- **Supporting Informed Technology Decisions**

With accurate, up-to-date data, architects can assess new tools based not on hype but on actual performance and suitability. This reduces risk and ensures that technology upgrades are evidence-based rather than speculative.

10.2.2: Collaborative Networks

Collaborative Networks transform individual lessons into collective knowledge:

- **Creating Knowledge-Sharing Spaces**

Regular sessions—such as architecture community calls, brown bags, or internal conferences—allow architects and engineers to share discoveries, lessons, and best practices. These forums foster a learning culture and enable the organization to evolve collectively.

- **Transforming Experience into Shared Learning**

The network captures insights from real-world project experiences and converts them into **generalized practices and reusable guidelines**. This practice ensures that one team's lessons benefit many others, increasing the organization's overall learning speed.

10.2.3: Operating Model

The **Operating Model** integrates learning directly into the organization's daily rhythm:

- **Learning as a Built-In Process**

Knowledge-sharing processes are not an afterthought; they are intentionally designed into the system. Teams are encouraged to document lessons learned and distribute them formally and informally regularly.

- **Systematic Capture and Reuse**

Insights gained from projects are captured, structured, and made accessible to others. These insights reduce redundant work, accelerate future decision-making, and ensure **continuous organizational improvement**.

11: Bonus: Leveraging Generative AI within Grounded Architecture Framework

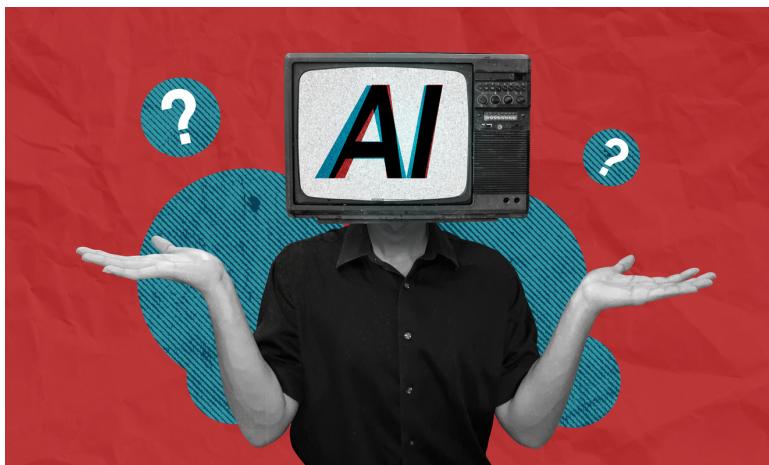


image by mininix doodle from istock

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.

Before concluding the section on the Grounded Architecture Framework, I also want to explore the **intersection of Generative AI (GenAI) and IT Architecture**, specifically through the lens of the **Grounded Architecture framework**.

This review is a **practical guide based on my real-world experiences** applying GenAI to architectural work. As someone who works actively within architectural practices in complex environments, I've experimented with GenAI tools to see how they can enhance our workflows. What follows are initial insights, lessons learned, and practical suggestions for other architects looking to do the same.

Generative AI has rapidly evolved from an experimental novelty into a

powerful tool capable of producing text, code, designs, and more. In architecture, where we often face fast-changing demands and **fragmented, slow-to-adapt tools**, GenAI shows promise for improving efficiency, automating repetitive tasks, and accelerating **data-informed decision-making**.

However, I've also observed how quickly this promise can lead to new challenges—such as increased complexity, unclear ownership, or ethical risks—if not used intentionally. This is where the **Grounded Architecture** approach has proven essential; it provides a clear framework for using GenAI **responsibly, adaptively, and strategically**.

This section aims to help IT and Enterprise Architects **get started with GenAI thoughtfully and pragmatically**. I'll cover:

- What GenAI can realistically accomplish today
- How it can be integrated into architectural practices
- Valuable use cases I've discovered
- The benefits and common pitfalls
- Best practices for responsible use
- Potential future developments

The goal is to help you **view GenAI as a strategic asset**, rather than just a passing trend. If utilized wisely, it can enhance—not replace—architectural thinking and decision-making.

An appendix includes a set of [GenAI prompts¹](#) I've tested for various tasks.

(True to the spirit of this work, the first version of this section was created through early exploration using the Gemini Deep Research chatbot, refined with ChatGPT, polished with Grammarly, and significantly edited to reflect my own experiences.)

¹[gen-ai-prompts](#)

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

Generative AI (GenAI) refers to a class of artificial intelligence systems that are trained on vast datasets to generate **new, original content**—including text, code, images, and designs—rather than simply analyzing or predicting based on existing data. While traditional AI excels at classification and pattern recognition, GenAI is designed for **creation and augmentation**.



image by lemon tm from istock

Users interact with GenAI through **natural language prompts**, making these systems both **accessible and highly flexible** for IT and Enterprise Architects navigating complex environments.

In my early hands-on experiences, GenAI has proven to be a valuable addition to architectural practice—enhancing how we **create, analyze, automate, communicate, and make decisions**.

Here's how GenAI can empower architects today:

11.1.1: Creating and Improving Content

GenAI significantly speeds up the creation of common architectural artifacts, such as:

- Drafting reports, technical documentation, meeting summaries, emails, and Architecture Decision Records (ADRs)
- Generating starter code, architecture diagrams, and design notes
- Rapidly iterating on written content or code, reducing cycle time

Large Language Models (LLMs) like GPT excel in this area, allowing architects to work faster with fewer blank slates.

11.1.2: Analyzing Data and Spotting Patterns

Architects often work with fragmented data sources. GenAI helps by:

- Synthesizing structured data (e.g., spreadsheets, databases) and unstructured data (e.g., PDFs, meeting notes)
- Identifying architectural **anti-patterns**, outdated components, and **technical debt**
- Translating visual models into structured data for easier analysis and comparison

These capabilities enable **faster assessments** of large systems and provide better visibility into architectural health.

11.1.3: Automating Repetitive Tasks

Many routine activities can be streamlined, including:

- Drafting documentation
- Conducting standard code reviews
- Extracting requirements from text documents
- Generating solution options or transition scenarios

These automations free architects to focus on **strategic and creative tasks**.

11.1.4: Enhancing Communication and Collaboration

With the rise of conversational AI interfaces, GenAI can serve as:

- An internal **architecture assistant**, capable of answering questions about standards, technologies, or system dependencies
- A tool for **explaining technical concepts** to non-technical stakeholders
- A medium for sharing knowledge across teams, even asynchronously

When used effectively, GenAI improves **clarity, accessibility, and collaboration** throughout the organization.

11.1.5: Offering Suggestions and Recommendations

GenAI can support architects in ideation by:

- Proposing architectural patterns or alternatives
- Recommending technologies based on known constraints
- Outlining modernization or migration strategies

These suggestions should be viewed as **starting points**, not conclusions—the architect remains the ultimate decision-maker.

11.1.6: The Role of Retrieval-Augmented Generation (RAG)

One of the critical enablers for applying GenAI in enterprise contexts is **Retrieval-Augmented Generation (RAG)**.

Standard GenAI models respond based solely on their training data. RAG enhances this by **searching for and retrieving relevant information** from real-time, trusted company sources—such as:

- Internal documentation

- Architecture repositories
- Technical standards and compliance databases

This allows the AI to **anchor its outputs in the most current, business-specific knowledge.**

RAG reduces inaccuracies, improves response quality, and ensures outputs are **contextual and aligned with organizational reality**—which is essential for data-informed decision-making and maintaining architectural integrity.

In summary, GenAI—especially when combined with RAG—has the potential to **enhance architectural practice at every level**, from daily efficiency to strategic insights. The next challenge is not *whether* to use GenAI but **how to utilize it effectively and responsibly**.

11.2: Integrating GenAI with Grounded Architecture

There is significant potential for **Generative AI (GenAI)** to enhance architectural practices when viewed through the **Grounded Architecture** framework. The benefits are especially evident in key areas such as **data analysis**, **knowledge sharing**, and **operational efficiency**—all core elements of Grounded Architecture.

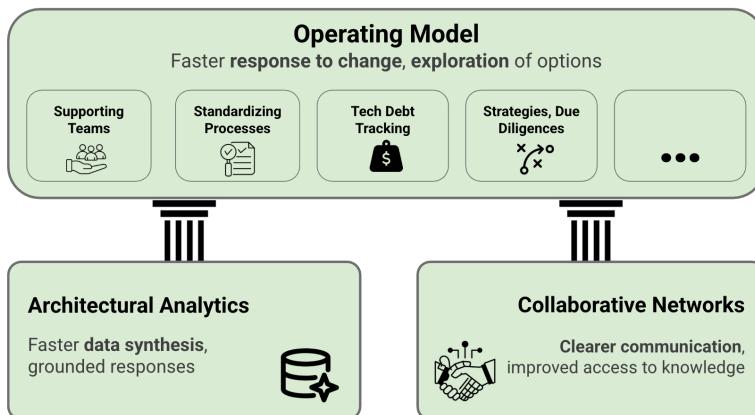


Figure 1: Potential applications of Generative AI within the Grounded Architecture Framework.

By incorporating GenAI capabilities into the three foundational pillars of Grounded Architecture—**Lightweight Architectural Analytics**, **Collaborative Networks**, and the **Operating Model**—we can achieve new levels of scalability, clarity, and speed. Below is a breakdown of opportunities within each pillar.

11.2.1: GenAI in Lightweight Architectural Analytics (Better Understanding the Current State)

- **Gathering and Processing Data**

GenAI can automate the extraction and summarization of informa-

tion from various sources—such as code repositories, cloud billing data, support tickets, and internal documentation. **Retrieval-Augmented Generation (RAG)** can help consolidate these into a unified and easily accessible context.

- **Recognizing Patterns and Anomalies**

Advanced models can assist in identifying recurring architectural patterns, spotting anomalies, uncovering security vulnerabilities, and flagging areas of technical debt—thus enhancing our analytical capabilities.

- **Generating Reports and Dashboards**

GenAI can support the automated creation of architectural reports and dashboards, making key insights more **timely, consistent, and widely accessible**.

11.2.2: GenAI in Collaborative Networks (Working Together More Effectively)

- **Managing and Sharing Knowledge**

Static repositories like Architecture Decision Records (ADRs) and design documents can be transformed into **dynamic, conversational knowledge bases**, improving discoverability and everyday usefulness.

- **Assisting with Communication**

Drafting messages, presentations, and stakeholder updates becomes more efficient and consistent, helping architects effectively communicate technical content to diverse audiences—from developers to executives.

- **Summarizing Meetings**

GenAI can capture and summarize discussions, decisions, and action points, reducing the risk of information loss and enhancing follow-through across teams.

11.2.3: GenAI in the Operating Model (Accelerating How Architecture Gets Done)

- **Supporting Code and Documentation**

AI-powered assistants can help architects and teams write, review,

or refine code and supporting documentation, especially for repeatable tasks or boilerplate content.

- **Generating Architectural Artifacts**

Initial drafts of diagrams, roadmaps, risk registers, or architecture descriptions can be co-created with GenAI, saving time and establishing a **clear starting point**.

- **Tracking and Prioritizing Technical Debt**

Automating the detection, documentation, and prioritization of technical debt can lead to faster remediation and more informed architectural trade-offs.

- **Accelerating Due Diligence**

During mergers, acquisitions, or audits, GenAI can summarize large volumes of technical material, helping leaders **quickly grasp risks and opportunities**.

- **Standardizing Governance and Practices**

Drafting architecture standards, policies, or review checklists using AI can lead to **greater consistency** and reduce the effort required to onboard new teams or scale practices.

- **Assisting in Strategy Development**

By analyzing existing data and external signals, GenAI can assist with initial drafts of technology strategy, future scenarios, or architectural vision documents.

GenAI, when aligned with the **Grounded Architecture framework**, has the potential to **transform architectural work**—not by replacing architects, but by **amplifying their capabilities**. Whether automating analysis, enhancing communication, or accelerating strategy development, GenAI can support more informed, efficient, and adaptive architectural practices.

As with any powerful tool, **intentional integration** is crucial. When grounded in solid data, connected people, and well-designed processes, GenAI becomes a valuable strategic asset that helps architects achieve **more, faster, and with greater clarity**.

11.3: Practical Examples: GenAI in Action

I view **Generative AI** as a valuable tool for architects. By **automating repetitive and time-consuming tasks**, it allows architects to concentrate on high-impact responsibilities—such as evaluating architectural trade-offs, collaborating with teams, interpreting outcomes, and applying sound judgment. These responsibilities are fundamental to the **Grounded Architecture** approach, and GenAI can enhance these workflows, making them **more efficient and scalable**.

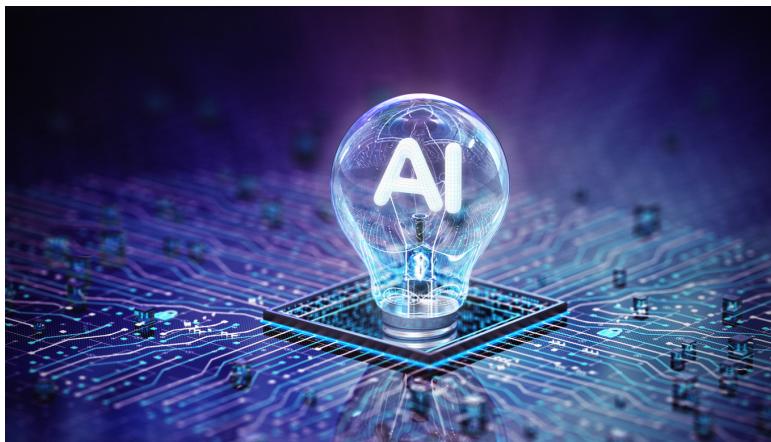


image by blackjack3d from istock

11.3.1: Use Case 1: Analyzing Dependencies and Technical Debt

- **Scenario:** Evaluating the dependencies among microservices and identifying outdated libraries or unnecessary couplings.
- **GenAI Application:** GenAI can analyze codebases, CI/CD data, and telemetry tools to uncover architectural problems. When combined with Retrieval-Augmented Generation (RAG), it can also examine internal documentation to highlight ownership, policy violations, or missing standards. Additionally, it may draft reports summarizing issues and offering recommendations.

- **Grounded Architecture Link:** This enhances **Lightweight Architectural Analytics** by increasing visibility into the current system health.

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

- **Scenario:** After a team selects a messaging technology, the rationale needs to be documented.
- **GenAI Application:** GenAI can summarize meeting notes and draft an ADR using templates, referencing relevant standards or prior decisions through RAG. Architects can then review, adjust, and finalize the document.
- **Grounded Architecture Link:** This supports an **Operating Model** (standardizing documentation) and **Collaborative Networks** (capturing and sharing decisions).

11.3.3: Use Case 3: Creating Architecture Diagrams from Descriptions

- **Scenario:** A team needs a quick system overview diagram for a strategy meeting.
- **GenAI Application:** AI diagramming tools can convert natural language descriptions into draft diagrams or generate visuals directly from code or configuration files. These drafts can then be refined for presentation purposes.
- **Grounded Architecture Link:** This aids collaboration and communication within **Collaborative Networks** and streamlines visuals within the **Operating Model**.

11.3.4: Use Case 4: Improving Requirements Analysis

- **Scenario:** Synthesizing user stories, emails, and transcripts into clear, actionable requirements.

- **GenAI Application:** NLP-based GenAI tools can extract, cluster, and organize raw input into coherent requirement lists, along with draft acceptance criteria and suggested test scenarios.
- **Grounded Architecture Link:** This supports early-phase design in the **Operating Model**, enhancing alignment between needs and solutions.

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

- **Scenario:** Designing a new recommendation engine while considering constraints like security and performance.
- **GenAI Application:** GenAI, supported by internal standards via RAG, can suggest patterns, generate outlines of options, and simulate trade-offs using historical benchmarks or current system data. Architects can use these insights to make informed final decisions.
- **Grounded Architecture Link:** This advances design and evaluation within the **Operating Model**, promoting data-driven decision-making.

11.3.6: Use Case 6: Enhancing Code Review Processes

- **Scenario:** Ensuring submitted code adheres to internal quality, security, and style guidelines.
- **GenAI Application:** Integrated into CI/CD, AI tools can flag issues, recommend fixes, summarize changes, and ensure consistency across reviews—supporting both automation and mentorship.
- **Grounded Architecture Link:** This reinforces quality and governance within the **Operating Model** and helps reduce technical debt.

These real-world examples demonstrate how GenAI can **complement the architect's role** rather than replace it. By streamlining workflows, enhancing analysis, and accelerating documentation, GenAI enables architects to focus on what truly matters: **strategic thinking, human collaboration, and informed decision-making**—the pillars of Grounded Architecture.

11.4: Balancing Benefits, Risks, and Challenges

Based on my early experiences, I believe that **Generative AI (GenAI)** can significantly enhance architectural work, especially when applied thoughtfully within the **Grounded Architecture** framework. But it's crucial to recognize that GenAI is not a panacea. It introduces new complexities and risks that must be managed carefully.

11.4.1: Key Benefits

The benefits of GenAI align closely with the principles of this framework:

- Efficiency corresponds with pragmatism.
- Faster data analysis supports data-informed decisions.
- Improved knowledge sharing strengthens collaborative networks.
- Better evaluation of options fosters adaptability.

More specifically, GenAI can:

- **Increase Efficiency and Productivity:**

GenAI automates routine tasks such as drafting documentation (e.g., SOPs and ADRs), creating diagrams, reviewing code, and analyzing system data. This automation frees up time for strategic work that requires deeper human insight.

- **Improve Consistency and Quality:**

AI-generated outputs can adhere to predefined formats and templates, enhancing compliance with architectural standards. In code reviews, this consistency helps reduce human errors and subjective variations.

- **Provide Faster, Data-Driven Decision Support:**

GenAI rapidly processes and synthesizes diverse data, thereby supporting Lightweight Architectural Analytics and uncovering insights that would otherwise take much longer to discover.

- **Enhance Collaboration and Knowledge Sharing:**
By utilizing RAG-powered chatbots, document summarization, and conversational interfaces, GenAI improves access to institutional knowledge and facilitates clearer communication across teams and audiences.
- **Foster Innovation:**
By taking over repetitive tasks, GenAI allows architects to devote more time to exploring design options, evaluating trade-offs, and identifying innovative solutions.
- **Make Architecture More Accessible:**
GenAI can help translate technical content for non-technical stakeholders through visualizations, summaries, or simplified explanations, supporting the objective of Grounded Architecture to democratize architectural thinking.

11.4.2: Key Risks and Challenges

Despite these promising benefits, there are limitations and risks that require careful management. To ensure that GenAI is effective and sustainable, **governance, oversight, and ethical awareness** are essential.

- **Accuracy and Reliability (AI “Hallucinations”):**
GenAI can produce confident but inaccurate or misleading outputs. Each result must be validated by a human, and this review process may offset time savings in critical scenarios.
- **Security and Data Privacy:**
Sharing sensitive data with cloud-based GenAI services poses a risk of **data leaks or unintended model training**. Implementing careful access control, anonymization, and encryption is critical for safeguarding data.
- **Bias and Ethical Concerns:**
AI models can amplify existing biases present in training data. Without intervention, this could lead to **unintended consequences** in architectural recommendations, hiring practices, or strategic decisions.
- **Intellectual Property and Copyright Uncertainty:**
The legal landscape surrounding AI-generated content is still evolving. AI outputs might inadvertently include material derived from

copyrighted training data, raising **questions about ownership and reuse**.

- **Need for Human Oversight:**

GenAI serves as a tool, not a replacement for human judgment. Architects must maintain control, applying their expertise, context awareness, and strategic thinking. Over-reliance on GenAI may lead to poor decisions.

- **Cost and Resource Demands:**

Implementing GenAI solutions—especially enterprise-grade or private models—requires **significant computing power, integration efforts, and financial investment**, both upfront and ongoing.

- **Performance and Latency:**

Complex prompts and models may respond slowly, particularly in real-time scenarios, impacting the **user experience** or delaying interactions where speed is critical.

- **Integration Complexity:**

Seamlessly incorporating GenAI into existing workflows and architectural tools involves a **non-trivial technical setup**, including managing APIs, data pipelines, prompt engineering, RAG configurations, and governance layers.

- **Model Limitations (Context Window and Memory):**

GenAI models have constraints on how much context they can handle simultaneously. In complex architectural tasks involving large systems or lengthy documents, these limitations can affect the output's usefulness.

GenAI presents a compelling opportunity to **enhance architectural effectiveness**, but it must be introduced with **awareness of its limitations**. Like any powerful tool, it has trade-offs. When implemented with responsible governance and integrated with human oversight, GenAI can serve as a **valuable partner to architects**, helping them navigate complexity, accelerate delivery, and foster continuous learning.

11.5: Making GenAI Work Within Grounded Architecture

As organizations increasingly incorporate Generative AI (GenAI) into their operations, it's essential to approach this transformative technology with **strategic intent, ethical awareness, and structured execution**. GenAI opens the door to new levels of **efficiency, innovation, and insight**, but it also introduces **challenges** around accuracy, governance, integration, and trust.



image by aree sarak from istock

To responsibly integrate GenAI in a way that **aligns with organizational goals and the principles of Grounded Architecture**, I've found the following practices especially valuable:

- **Start with Clear Goals and Prioritized Use Cases** Focus on high-value, low-risk applications aligned with strategic priorities. Use pilot projects to build confidence and refine your approach.
- **Build Strong Data Governance** High-quality, secure, and well-managed data is foundational—especially for Retrieval-Augmented

Generation (RAG). Prioritize lineage, privacy, encryption, and role-based access.

- **Embed Human Oversight (Human-in-the-Loop)** Architects and decision-makers must remain central. Always include checkpoints for human validation, and define accountability clearly to avoid automation bias.
- **Adopt Ethical and Principle-Based Governance** Guide your AI efforts with principles like **fairness, reliability, privacy, inclusiveness, and accountability**. Consider creating an internal AI review board or Center of Excellence.
- **Leverage Grounding with RAG for Contextual Accuracy** To minimize hallucinations and increase business relevance, connect GenAI systems to internal, curated knowledge. Make RAG a core part of your enterprise AI design.
- **Make Deliberate Build-vs-Buy Choices** Evaluate whether to use off-the-shelf tools with built-in AI features, consume public APIs, or build custom AI stacks. Factor in complexity, cost, and long-term adaptability.
- **Design Secure, Modular AI Architectures** Apply **Security by Design** principles. Limit permissions, use identity-based access, and create modular systems that are easier to scale, monitor, and govern.
- **Test, Monitor, and Continuously Improve** Treat GenAI like a living system. Monitor for **performance drift, accuracy, cost (AI FinOps), and ethical risks**. Create feedback loops and evolve the system iteratively.
- **Promote AI Literacy and Responsible Use** Educate users on how GenAI works, its strengths and limitations, and how to engage critically with its output. Foster a culture that values **curiosity, caution, and accountability**.

In conclusion, **Grounded Architecture** provides a strong foundation for GenAI adoption. Organizations that are **data-driven, collaborative, and pragmatic** are uniquely positioned to harness the potential of GenAI—without compromising trust, alignment, or architectural integrity.

Success depends not just on tools, but on **clear goals, strong governance, human judgment, and a culture of continuous learning and ethical responsibility**.

11.6: The Evolving Landscape

As Generative AI (GenAI) and other AI technologies continue to mature, we are entering a phase of **accelerated architectural evolution**. These advancements promise **faster feedback loops, real-time adaptability, and smarter systems**—all of which align with and enhance the principles of Grounded Architecture.



image by galeanu mihai from istock

These trends suggest a future where Enterprise Architecture (EA) becomes not just reactive but also proactive, adaptive, and continuously informed by AI systems.

11.6.1: Real-Time, Augmented Enterprise Architecture

EA may transition from static, periodic documentation to a “**living architecture**” that is continuously updated through AI monitoring and automated insights.

- AI agents could detect changes in infrastructure, systems, and usage patterns, instantly reflecting them in architectural models or knowledge graphs.

- Architects will become **augmented decision-makers**, utilizing GenAI as a “copilot” to explore trade-offs, simulate impacts, and prioritize initiatives in real time.
- The Lightweight Architectural Analytics repository could evolve into an **“operating system for change”**, enabling organizations to respond faster and smarter.

11.6.2: Rise of Agentic AI

More autonomous AI agents will begin to handle **complex, multi-step tasks** such as planning, tool usage, and workflow optimization.

- In EA, this could mean **continuous governance checks, detection of architectural drift, and self-healing systems**.
- While this brings the vision of **self-optimizing organizations** closer to reality, it introduces critical concerns regarding **trust, safety, and control**.

11.6.3: Digital Twins of Organizations (DTOs)

DTOs create **real-time, data-rich replicas** of an enterprise’s systems, processes, and operations.

- Paired with GenAI, DTOs enable **predictive modeling, scenario testing, and real-time optimization**.
- This aligns closely with Grounded Architecture’s goal of maintaining a **current and comprehensive view** of the enterprise.
- Leading organizations like **BMW** and **UPS** are already investing in this approach.

11.6.4: Increased Democratization and Collaboration

AI tools featuring **natural language interfaces**, such as chatbots and auto-generated visualizations, will **expand access to architectural insights** across the organization.

- Business users will be able to query EA repositories conversationally.
- AI-generated reports and explanations will help bridge the gap between technical and non-technical stakeholders.
- These tools will strengthen **Collaborative Networks** by fostering **shared understanding** and **inclusive decision-making**.

11.6.5: Composable and Modular AI Architectures

As innovation and experimentation accelerate, future AI systems will emphasize **modularity** and **interoperability**.

- Organizations will combine various components, such as LLMs, vector databases, RAG components, and AI agents.
- This aligns directly with Grounded Architecture's emphasis on **adaptability**, enabling architectures to evolve without requiring complete rewrites.

11.6.6: The Evolving Role of the Architect

The architect's role will further shift towards **strategic enablement**, with responsibilities such as:

- Governing AI adoption and ethics
- Curating enterprise data and model quality
- Ensuring alignment between AI outputs and business context
- Facilitating collaboration across domains
- Critically evaluating AI suggestions before implementation

New roles—such as **Enterprise AI Architect**—may emerge, focusing on designing and governing the intersection of architecture, AI, and business strategy.

11.6.7: Vertical AI Specialization

Industry-specific AI solutions (e.g., for healthcare, finance, and logistics) will continue to grow.

- Architects will need a **strong understanding of domain-specific constraints and opportunities** to guide responsible AI adoption.
- This reinforces the necessity for **context-aware architecture practices**, which Grounded Architecture directly supports.

The future of architecture will be **faster, smarter, and more collaborative**, significantly influenced by GenAI and emerging AI capabilities. Grounded Architecture provides a foundation that is **well-suited for this transformation**, anchored in adaptability, transparency, and strategic judgment.

As the landscape evolves, architects have the opportunity not just to **adapt**, but to **lead**—curating AI's role responsibly while driving meaningful business outcomes through thoughtful design and governance.

11.7: Final Thoughts

As **Generative AI (GenAI)** evolves from a novel concept to a powerful enterprise tool, architects have a significant opportunity to **transform their practice** for greater speed, adaptability, and strategic impact.

Architectural work often faces challenges due to fragmented tools, manual processes, and an overreliance on meetings and subjective opinion. GenAI has the potential to:

- **Automate repetitive tasks**
- **Accelerate data analysis and documentation**
- **Enhance collaboration and communication**
- **Facilitate faster, more informed decision-making**

However, a careless approach to implementing GenAI can also introduce risks, including inaccurate outputs, ethical concerns, integration difficulties, and hidden costs.

Grounded in real use cases and empirical experimentation, this report outlines how GenAI can empower architects to:

- **Operate at scale** by minimizing time spent on low-value tasks
- **Increase adaptability** with tools that evolve alongside the organization
- **Enhance decision-making** through the synthesis and analysis of complex data
- **Improve alignment** by providing clear and timely architectural insights
- **Foster continuous learning** via automated exploration and structured feedback loops

GenAI integrates seamlessly with the **three foundational pillars** of the Grounded Architecture framework:

- **Lightweight Architectural Analytics:** Automate the gathering of insights, pattern detection, and reporting.
- **Collaborative Networks:** Facilitate knowledge sharing through AI-driven summaries, chatbots, and communication support.

- **Operating Model:** Aid in creating architectural artifacts, maintaining standards, tracking technical debt, and shaping strategic direction.

Generative AI is more than just a tool; it is a catalyst for architectural evolution. With the Grounded Architecture framework as a guide, architects can leverage AI's potential to **enhance, rather than replace, their expertise**, leading their organizations into a new era of intelligent and adaptable enterprise design.

11.8: To Probe Further

- Generative AI References²
- Generative AI Prompts³

²<https://grounded-architecture.io/gen-ai-references>

³<https://grounded-architecture.io/gen-ai-prompts>

11.9: 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?*

11.10: Tables

11.10.1: 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 daily work, like 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.

11.10.2: GenAI Best Practices 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.