



Grounded Architecture

Redefining IT Architecture Practice in the Digital Enterprise

Željko Obrenović

Grounded Architecture

Redefining IT Architecture Practice in the
Digital Enterprise

Željko Obrenović

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

This version was published on 2025-05-20



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	3
1.2: What Will You Learn?	7
1.3: Beyond Rigid Frameworks: IT Architecture As an Adaptable Outcome-Oriented Practice	11
1.4: Key Influences	14
1.5: Why This Book?	15
1.6: A Part of the Bigger Picture: A Trilogy in Four Parts	16
1.7: A Bit of Personal History	18
1.8: The Structure of the Book	19
1.9: Stay Connected	20
1.10: Acknowledgments	21
2: Context: Fast-Moving Global Organizations	22
2.1: Global Scale	24
2.2: Multi-Dimensional Diversity	26
2.3: Nonlinear Growth Dynamics	29
2.4: Synergy and Transformation Pressures	31
2.5: Decentralization and Loose Coupling	33
2.6: Questions to Consider	35
3: Goals: Adapting, Growing, and Using Data	36
3.1: Goal 1: Executing At Scale	38
3.2: Goal 2: Adaptivity	39
3.3: Goal 3: Increasing Quality of Decisions with Data	40
3.4: Goal 4: Maximizing Organizational Alignment	41
3.5: Goal 5: Maximizing Organizational Learning	42
3.6: Questions to Consider	43

CONTENTS

Part I: Grounded Architecture Framework	44
4: Grounded Architecture Framework: Introduction	45
5: Lightweight Architectural Analytics	49
5.1: Examples of Lightweight Architectural Analytics Tools	52
5.2: Requirements For A Lightweight Architectural Analytics	61
5.3: Building Lightweight Architectural Analytics	63
5.4: Using Lightweight Architectural Analytics	65
5.5: To Probe Further	67
5.6: Questions to Consider	68
6: Collaborative Networks	69
6.1: Background: Central vs. Federated Architecture Practice	72
6.2: The Hybrid Model	74
6.3: Building Collaborative Networks	78
6.4: To Probe Further	80
6.5: Questions to Consider	81
7: Operating Model: General Principles	82
7.1: Examples of Architecture Activities	85
7.2: Guiding Principles for Architectural Excellence: Policies, Autonomy, and Engagement	88
7.3: Embracing Diversity	95
7.4: Setting Boundaries	97
7.5: To Probe Further	100
7.6: Questions to Consider	101
8: Cooperation-Based Operating Model: Six Simple Rules	102
8.1: Background: Hard and Soft Management	104
8.2: Collaboration Approach	108
8.3: Rule 1: Understand What Your People Do	110
8.4: Rule 2: Reinforce Integrators	112
8.5: Rule 3: Increase the Total Quantity of Power	114
8.6: Rule 4: Increase Reciprocity	116
8.7: Rule 5: Extend the Shadow of the Future	118
8.8: Rule 6: Reward Those Who Cooperate	120
8.9: To Probe Further	122
8.10: Questions to Consider	123

9: Operating Model: Nudge, Taxation, Mandates	124
9.1: Nudging	127
9.2: Taxation (Economic Incentives)	129
9.3: Mandates and Bans	131
9.4: Questions to Consider	133
10: Transforming Organizations with Grounded Architecture	134
10.1: Executing at Scale	136
10.2: Adaptivity	139
10.3: Improving the Quality of Decision-Making with Data	142
10.4: Maximizing Organizational Alignment	145
10.5: Maximizing Organizational Learning	147
10.6: Questions to Consider	150
11: Next Steps: Leveraging The Potential of Generative AI	151
11.1: What Generative AI Can Do for Today's Architect	154
11.2: Integrating GenAI with Grounded Architecture: Connecting Capabilities to the Framework	156
11.3: Practical Examples: GenAI in Action	159
11.4: Balancing Benefits, Risks, and Challenges	163
11.5: Best Practices for Responsibly Adopting GenAI	167
11.6: The Evolving Landscape	170
11.7: Questions to Consider	173
11.8: Tables	174
11.9: To Probe Further: References	179

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 stuck in an ivory tower**? Look no further! This book will share an approach to running an IT architecture practice in larger organizations based on my experience at AVIV Group, eBay Classifieds, and Adevinta. I call it “Grounded Architecture”—because it’s all about having your feet on the ground and your architecture **firmly planted in reality**.

The approach presented in this book connects architecture to every nook and cranny of your organization, making sure you’re not a lone wizard in a tower casting spells that no one understands. Grounded Architecture prioritizes connections with people and data over fancy theories, processes, and tools. The books will offer you new insight into IT architecture, redefining it as an **adaptable** and **outcome-oriented** practice.

The name “Grounded Architecture” is a hat-tip to the **Grounded Theory**¹ methodology. Grounded Theory is about developing theories from the data you gather from the real world rather than pulling them out of thin air. Likewise, Grounded Architecture evolves from honest feedback and data, not from dusty old books of abstract principles.

¹https://en.wikipedia.org/wiki/Grounded_theory

1.1: Grounded Architecture Overview

This book breaks down Grounded Architecture into two main parts (Figure 1):

- **Framework:** The nuts and bolts you need to get your Grounded Architecture practice up and running.
- **Guiding Principles:** Handy tips and inspirations to help you bring these ideas to life.

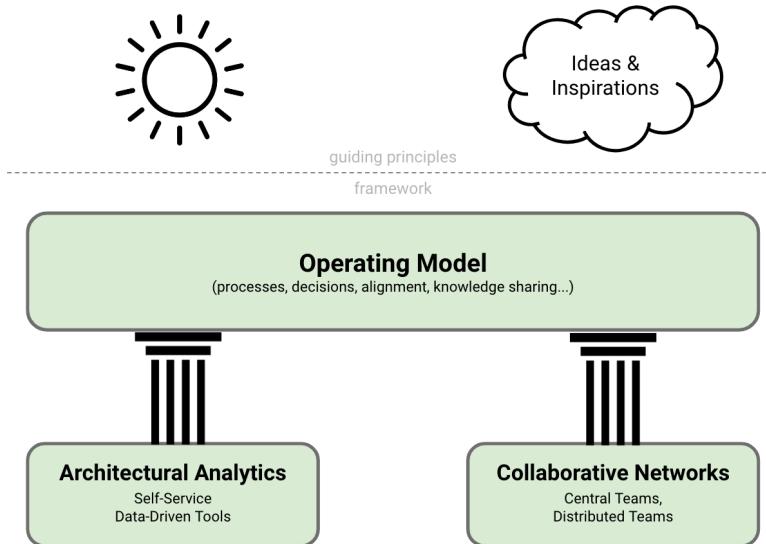


Figure 1: Grounded Architecture Overview.

Grounded Architecture framework consists of three elements:

- Lightweight Architectural Analytics (Data),
- Collaborative Networks (People),
- Operating Model.

Lightweight Architectural Analytics keeps your architecture decisions based on up-to-date and complete information about your organization's tech landscape.

Collaborative Networks represents a strong network of people working on architecture across your organization, ensuring your efforts make a difference.

Lastly, the sections *Operating Model: General Principles*, *Cooperation-Based Operating Model*, *Leveraging The Full Potential of Generative AI for IT Architecture*², and *IT Governance: Nudge, Taxation, Mandates* define a collection of processes and agreements that lets architects do their thing, leveraging data and people connections to create a powerful, organization-wide impact.

As part of my Grounded Architecture quest, I've devised several guiding principles and tools to help you sprinkle Grounded Architecture magic in your practice. While traditional IT architecture literature often zeros in on technical know-how, this collection dares you to think differently about IT architecture and to dive into the rich insights of social, behavioral, and management sciences. These resources are grouped into several parts and are sure to make your journey both insightful and entertaining:

- **On Being an Architect:**

- Building Skills
- Making Impact
- Leadership
- Thinking Like an Architect: Architects as Superglue
- Balancing Curiosity, Doubt, Vision, and Skepticism
- Architects' Career Paths

- **On Human Complexity:**

- The Culture Map: Architects' Culture Compass
- The Human Side of Decision-Making
- Effortless Architecture

- **On Strategy³:**

²gen-ai-potential

³strategy

- Enterprise Architecture as Strategy⁴
- Hunting the Ghost of Business Architecture⁵
- Outsourcing Strategies⁶
- Achieving Market Leadership⁷
- Value-Based Strategy⁸
- Marketplace Strategies: What Digital Marketplaces Sell?⁹
- Connecting Marketing, Sales, and Customer Service Strategies¹⁰
- Culture As a Strategy (aka Culture Eats Strategy for Breakfast)¹¹

- **Expanding the Architect’s Toolkit: Learning From Other Fields:**

- Economic Modeling With ROI and Financial Options: Learning From the Finance Field
- Architecture in Product-Led Organizations: Learning From Customer-Centric Fields
- Decision Intelligence in IT Architecture: Learning From Data, Social, and Managerial Fields
- How Big Transformations Get Done: Learning From Mega-Projects

If all this is not enough, in the online appendix I outline several resources that have proven vital in my journey as an IT architect. I routinely refer back to them and carry them in my practitioner “backpack,” so I added their summaries in the online [appendix](#), including:

- Favorite Quotes¹²
- Bookshelf¹³
- Resources for Managing, Growing, and Hiring Architects¹⁴

⁴[ea-as-strategy](#)

⁵[business-architecture](#)

⁶[outsourcing](#)

⁷[market-leadership](#)

⁸[value-based-strategy](#)

⁹[marketplaces](#)

¹⁰[marketing-sales-strategy](#)

¹¹[culture-strategy](#)

¹²<https://grounded-architecture.io/quotes>

¹³<https://grounded-architecture.io/bookshelf>

¹⁴<https://grounded-architecture.io/growing>

- Resources for Effective Communication¹⁵
- Resources for Working With Toxic Colleagues¹⁶
- Resources for Dealing With Scapegoating at Work¹⁷
- EIC/ISO 25010 Standard¹⁸
- Cloud Design Patterns¹⁹

The rest of this book will explain the Grounded Architecture approach in detail. In this section, I want to share a few things about my motivation to write this book.

¹⁵<https://grounded-architecture.io/communication>

¹⁶<https://grounded-architecture.io/toxic-colleagues>

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

¹⁸<https://grounded-architecture.io/iso25010>

¹⁹<https://grounded-architecture.io/cloud-design-patterns>

1.2: What Will You Learn?

This book serves as a guideline to build and operate a **robust IT architecture practice**, ensuring alignment with organizational goals while adapting to the challenges of modern IT environments.

1.2.1: Key Topics

The book will provide you with tips and inspirations about work that you need to set up and run a modern IT architecture practice, including:

- Create flexible organizational and technical **structures** to support IT architecture work,
- Define **IT architecture roles** and responsibilities, skills, and career paths,
- Operate an **effective IT architecture practice** in complex multi-cultural organizations.

I also provide a number of other ad-hoc lessons and inspirations from my work.

1.2.2: Format

I have organized my lessons and insights in a form that, if you recognize the problems and are inspired by solutions, could use as a **high-level “playbook”** about how to work as an architect or run an architecture practice. I also provide more concrete tips on each discussed topic, finishing each section with questions you should consider when addressing these topics.

I invite you to read this book from **beginning to end**, following the progression from data and basic structures to management and organizational topics. However, you can also **browse the text** and start reading whatever interests you. I use many **illustrations** to create easy-to-remember pictures that you can associate with discussed topics, making the book usable across your organization as a **coffee table book**, serving as an inspiration, or sparking discussions.

1.2.3: Content

This book is **not technical**. We will not discuss the details of public cloud design patterns, security, reliability, how to optimize computer loads, or select the proper data storage. As a modern architect, you will need these skills, but there are already many great resources. This book is about **expanding your horizons** to apply your technical skills in complex organizations. You can broaden your horizons as a head or manager of architects and organize and support architects to use their technical skills more effectively as a team.

1.2.4: Is This A Proven Method?

Like with many similar books, you may be disappointed if you are looking for a scientifically proven “method” of running a modern architecture practice. This book is **personal and opinionated**, building on my daily experiences as an architect and head of an architecture practice. While subjective, this book can provide valuable insights for IT architects, their managers, and people working with architects. I have successfully **applied my approach in three different companies**, which gives it some generality and repeatability.

I invite others to share the lessons they have learned similarly. Even if opinionated and limited in scope, such practical reflections based on concrete examples have much more value for practitioners than abstract debates, formal methods, or academic analyses.

1.2.5: Who Should Read This Book?

When writing this book, I had a broad audience in mind. The articles should be helpful to both technical and non-technical people. The book can help IT architects better understand their value and place in a broader organization. I also hope the articles show the wider audience the benefits of staying close to and well-connected with architects.

1.2.6: Applying Ideas In Practice

I approach designing an architecture practice in complex organizations as an **art of cooking**. A modern IT architecture practice is like a master

chef's kitchen. As you will always work with many local ingredients, you can't just copy a recipe from one restaurant to another and expect it to taste the same.

This book shares some tasty tips and savory secrets on "cooking" up a thriving an architecture practice. But remember, **every kitchen is different**. Some ingredients and recipes are essential, while others might be left on the shelf. And don't be surprised if you need to add a pinch of something special from other sources to spice things up.



image by hispanolistic from istock

When I start an architecture practice, I see myself as a chef stepping into a new restaurant with a suitcase full of favorite spices. I blend in core preferred elements and foundational frameworks like a chef using their trusty spices and kitchen tools. But the **authentic flavor**—the magic that makes a dish unforgettable—comes from the **local ingredients**. In an organization, these local ingredients are the people: the in-house talent and culture that give the enterprise its unique taste.

While fundamental structures and practices may be consistent across diverse businesses, the most crucial elements—akin to fresh, **local ingredients** in cooking—you must procure locally. The staff's skills, experiences, and insights are the secret ingredients that personalize and refine the

architecture to align with specific business needs and goals. They are what render the architecture truly unique and effective.

1.3: Beyond Rigid Frameworks: IT Architecture As an Adaptable Outcome-Oriented Practice

In this book, I will use the terms **IT architecture** and **IT architects** in a broad sense. I don't adhere to strict divisions between enterprise, solution, technical, or software architecture, nor do I narrowly define roles such as enterprise architects, solution architects, or Staff+ technical leaders like principal engineers. Instead, I view IT architecture as a **cohesive function**—a bridge connecting technology, business, product, and organizational strategy. IT architecture is more than a set of duties; it serves as essential connective tissue, aligning these domains to work harmoniously.

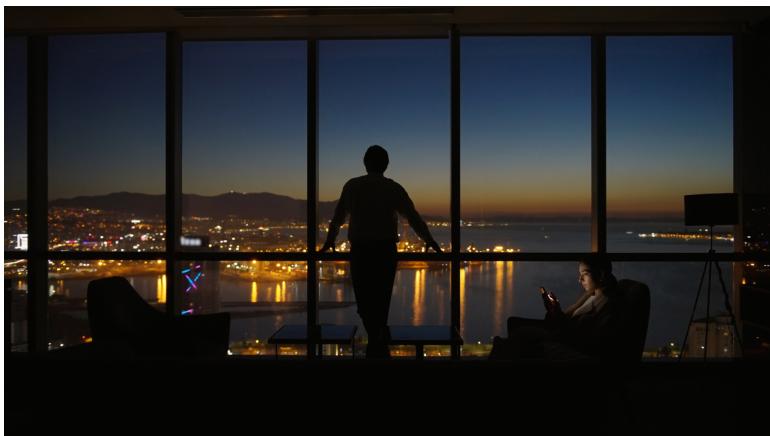


image by brightstars from istock

1.3.1: Grounding and Empowerment

Rather than advocating for rigid role distinctions, I propose **rethinking and empowering these roles** with an **outcome-oriented, data-informed perspective**, where each role is adaptable and focused on delivering value and impactful outcomes that align with the organization's

broader goals. This book emphasizes **grounding** your architecture in the organization with data, people connections, and pragmatic operating principles. You do not necessarily need to change your architecture team or roles to benefit from this grounding.

I believe IT architecture **isn't one-size-fits-all**; it adapts to each company's context, spanning strategic guidance, system integration, solution design, technical implementation, or hands-on leadership. This **adaptability** makes architecture both **powerful and relevant**, enabling architects to align with business goals while remaining responsive to technological change.

In some organizations, architecture may emphasize **business alignment**, ensuring technology investments support the mission. In others, it might focus on **technical depth**, where architects closely collaborate with engineering to build robust, scalable systems. Despite these variations, the core function remains: architecture guides coherence in complex ecosystems. For this reason, I consider IT architects anyone performing this bridging work, regardless of their title.

Focusing on architecture as an **integrative and adaptable function** allows us to appreciate the unique contributions of each architectural role. Rather than defining roles rigidly, this approach highlights architecture as a continuous, evolving practice that balances immediate technical needs with **long-term strategic vision**. This perspective empowers architecture to drive sustainable growth, adaptability, and innovation across the business.

1.3.2: Accepting Challenges

This broader view of IT architecture, while robust and integrative, presents unique challenges. Unlike traditional frameworks with well-defined roles and responsibilities, my perspective **requires constant adaptation** to evolving organizational needs. Traditional frameworks bring clarity and stability, outlining roles and providing structured pathways, making it easier to align efforts across a project's lifecycle.

In contrast, an adaptive, integrative view of architecture is **less rigid** and **more attuned** to a company's specific, shifting needs. This flexibility allows architects to bridge technology, business, product, and strategy holistically, responsive to change. However, IT architects and technical

leaders must continually **evaluate and recalibrate** their approach. This evolution might mean adjusting priorities, balancing high-level strategy with hands-on guidance, and frequently engaging with stakeholders to ensure alignment.

I believe deeply in this approach, though I acknowledge its demands. It calls for **insight** into business and technology, a readiness to **embrace ambiguity**, and the ability to make decisions amid shifting priorities. IT architects working within this model need a strong grasp of the company's vision and an **acute sense of timing**—knowing when to emphasize strategy or technical details. It requires agility, resilience, and the willingness to pivot as needs shift and to challenge assumptions to serve the business.

While challenging, this adaptive model of architecture offers rewards that traditional frameworks may lack. By not confining architects to strict roles, it enables solutions genuinely aligned with the organization's mission, fostering closer collaboration and responsiveness to new opportunities or challenges. This flexibility ultimately allows architecture to drive lasting value and growth—creating a foundation that supports both current goals and future innovation.

1.4: Key Influences

The Grounded Architecture approach also builds on many ideas that others have successfully used. Gregor Hohpe's **Architecture Elevator**²⁰ view of architecture has heavily inspired my work. In many ways, my work reflects the lessons learned from implementing Gregor's ideas in practice. Gregor described modern architects' functions as aligning organization and technology, reducing friction, and charting transformation journeys. Such modern architects ride the Architect Elevator from the penthouse, where the business strategy is set, to the engine room, where engineers implement enabling technologies.

In my quest to define modern architectural roles, I used Staff+ Engineering jobs as an inspiration for the development of architects. Tanya Reilly's book **The Staff Engineer's Path**²¹ and Will Larson's book **Staff Engineer: Leadership beyond the management track**²² are helpful guides in defining the responsibilities of modern architects. Overall, the Staff-plus engineering roles provide excellent examples for the development of architects.

Many other sources have influenced my work. You can find them in the online [Bookshelf appendix](#)²³.

²⁰<https://architectelevator.com/>

²¹<https://www.oreilly.com/library/view/the-staff-engineers/9781098118723/>

²²<https://staffeng.com/guides/staff-archetypes/>

²³<https://grounded-architecture.io/bookshelf>

1.5: Why This Book?

This book **generalizes my experiences** in written form. I have written these articles for several reasons. Firstly, writing helps me clarify and improve my ideas (Figure 2). As Gregor Hohpe once noted, every sentence you write frees up brain cells to learn new things.

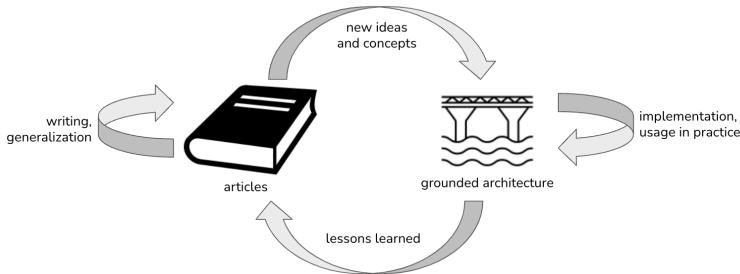


Figure 2: Writing a book helped me organize ideas, obtain new insights, improve principles and tools, and share the lessons learned.

I also needed this book for the **education of architects** and to **increase awareness about modern an architecture practices** in organizations I worked in. Having written content can significantly help to spread the message. As nicely described by Hohpe written word has distinct advantages over the spoken word:

- it scales: you can address a broad audience without gathering them all in one (virtual) room at the same time
- it's fast to process: people read 2-3 times faster than they can listen
- it can be easily searched or versioned.

Lastly, by generalizing and putting my experiences on paper, I aim to create more **usable materials to help others** in similar situations. I also expect helpful feedback from a broader community.

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

This book is a part of the collection of open-source tools and resources²⁴ I have built to help me in architectural work (Figure 3).

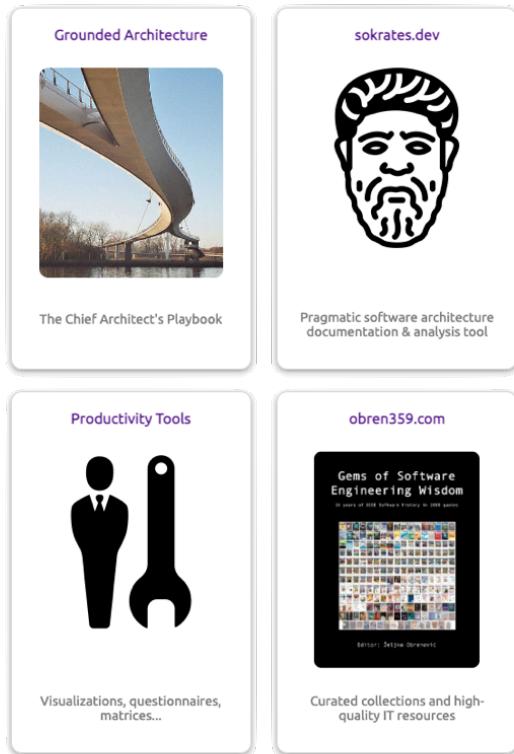


Figure 3: Grounded Architecture is a part of the collection of open-source tools and resources I have built in the past ten years to help me in architectural work.

²⁴<https://obren.io/>

The other resources include:

- **Sokrates**²⁵: an open-source polyglot source code analysis tool. Sokrates provides a pragmatic, inexpensive way to extract rich data from source code repositories. Sokrates can help you understand your code by making the size, complexity, and coupling of software, people interactions, and team topologies visible.
- **Productivity Tools**²⁶: A collection of more than 100 online tools I built to help me in my daily work.
- **359° Overview of Tech Trends**²⁷ is my collection of knowledge resources with podcasts and videos from over 20 authoritative, high-quality sources (IEEE, ACM, GOTO Conf, SE Radio, Martin Fowler's site, Ph.D. Theses). Architects need to learn fast, and finding good knowledge sources is difficult.

You can find more details about these tools on my homepage obren.io²⁸.

²⁵<https://sokrates.dev>

²⁶<https://obren.io/tools>

²⁷<https://www.obren359.com/>

²⁸<https://obren.io/>

1.7: A Bit of Personal History

The work presented in this book builds on **several years of my experience**. Most of this work originates from my current work as a **Chief Architect** at AVIV Group and previous works as a Principal Architect for eBay Classifieds and Adeventa.

Another vital part of my experience that shaped this book was my earlier experience as a **consultant and analyst** at the **Software Improvement Group**²⁹, where I've learned the value and pragmatics of data-informed decision-making. As a spin-off of this work, I've also built a tool called **Sokrates**³⁰, which enables efficient and pragmatic extraction of data about technology and organization from source code. This work has directly influenced my view of **Lightweight Architectural Analytics** part of the Grounded Architecture framework.

My experience as a **CTO** of **Incision**³¹, a startup, has helped me better understand the challenges of creating and running an IT organization.

My experience as a **researcher** at **Dutch Center for Computer Science and Mathematics (CWI)**³² and **Eindhoven Technical University (TU/e)**³³ provided me with a valuable background to do rigorous data analyses and research. From this research period, I want to highlight a collection of essays, **Design Instability**³⁴, that I co-authored with Erik Stolterman, where we connected experiences of designing/architecting in three disciplines: classical design, UX design, and software engineering. This work has helped me better relate to and learn from non-technical fields, a vital part of IT architects' job.

Lastly, my hands-on experience as a **software developer** has proven invaluable for my work as an architect.

²⁹<https://www.softwareimprovementgroup.com/>

³⁰<https://sokrates.dev>

³¹<https://incision.care>

³²<https://www.cwi.nl/en/>

³³<https://www.tue.nl/en/>

³⁴<https://design-instability.com/>

1.8: The Structure of the Book

I have organized the book into several main parts. In the introductory part, I describe the context in which my ideas have developed.

In the second part, I discuss the Grounded Architecture framework and describe its three elements: Lightweight Architectural Analytics, Collaborative Networks, and the Operating Model.

In the third part, I discuss the Guiding Principles of Grounded Architecture, grouped into several sub-parts: On Being an Architect, On Human Complexity, On Strategy, and Expanding the Architect's Toolkit.

Finally, I conclude with a summary and encourage you to explore further with the help of the plentiful external resources I've provided.

1.9: Stay Connected

You can find additional resources online at:

- <https://grounded-architecture.io>³⁵

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

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

³⁵<https://grounded-architecture.io/>

1.10: Acknowledgments

I want to thank all AVIV Group's Architecture Center of Excellence members and eBay Classifieds Virtual Architecture Team (VAT) members, who gave me invaluable feedback and discussions. Lastly, thank Peter Maas and Brent McLean for sponsoring and pushing for the development of data-informed architecture in our organizations.

The cover image is a photo of Nesciobrug³⁶. Credit: the botster, CC BY-SA 2.0, via Wikimedia Commons.



image by henk monster cc by 3 0 via wikimedia commons

³⁶https://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: Increasing Quality of Decisions 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: Increasing Quality of Decisions with Data

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

Some of the success criteria for this goal included:

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

3.4: Goal 4: Maximizing Organizational Alignment

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

Some of the success criteria for this goal included:

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

3.5: Goal 5: Maximizing Organizational Learning

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

Some of the success criteria for this goal included:

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

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

3.6: Questions to Consider

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

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

Part I: Grounded Architecture Framework

4: Grounded Architecture Framework: Introduction

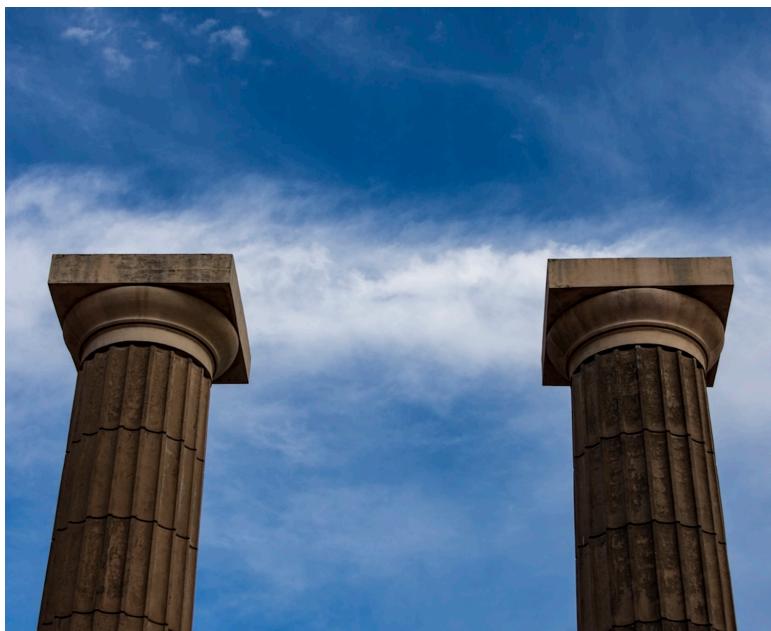


image by dario truco from istock

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 part, I will briefly introduce the Grounded Architecture framework. I chose the term “Grounded Architecture” to highlight that the primary goal of my approach is **avoid having an “ivory tower” architecture practice** disconnected from the organization. This disconnection is a real danger in a **fast-moving, global, and diverse setting**.

The Grounded Architecture framework is designed to be a **practical solution**. It aims to be **deeply rooted** in the organization, prioritizing **people’s interactions** and **data** over processes and tools. The goal is to connect an architecture practice to all parts and levels of the organization, serving as a practical antidote to the ‘ivory tower’ architecture. In this part of the ebook, I will delve into the framework’s structure and how it can be applied in real-world scenarios.

Grounded Architecture framework is an approach to setting organizational structures for an architecture practice, and it has three elements:

- Lightweight Architectural Analytics (Data),
- Collaborative Networks (People),
- The Operating Model.

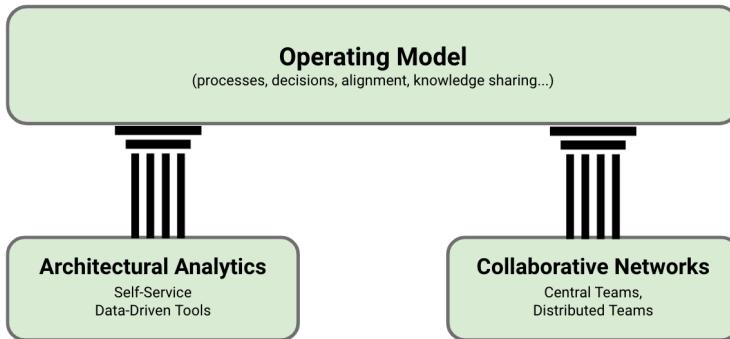


Figure 1: The Grounded Architecture framework.

Lightweight Architectural Analytics is a **system of tools and resources** that enables architects to make **data-informed decisions** based on a real-time and complete overview of the organization's technology landscape. The [Lightweight Architectural Analytics section](#) provides more details.

Collaborative Networks connect all the people doing architecture across the organization. These networks are crucial to ensure that an architecture practice has any **tangible impact**. The [Collaborative Networks section](#) provides more details.

Lastly, the *Operating Model* introduces:

- **General Principles:** I've defined several **general guidelines** on directing architects to do everything an architecture practice typically does, leveraging Lightweight Architectural Analytics and Collaborative Networks to create a data-informed, organization-wide impact.
- **Cooperation-Based Operating Model: Six Simple Rules:** Grounded Architecture is particularly well-suited for organizations organized based on cooperation. As businesses grow increasingly complex, the importance of cooperation within organizational structures cannot be overstated. The "Six Simple Rules" framework emphasizes that successful organizations leverage their members' collective judgment and cooperation. By fostering a culture of collaboration, organizations can better utilize their capabilities to address and resolve complex problems.

- **IT Governance: Nudge, Taxation, Mandates:** Governance in technology is a critical aspect of any operating model. I propose a governance model incorporating three styles: nudging, taxation, and mandates. Nudging involves subtle policy shifts that guide behavior without restricting choices, while taxation uses economic incentives to encourage desired outcomes. Mandates and bans provide direct regulations to ensure compliance and standardization. This multifaceted approach allows for flexible and effective governance, balancing innovation with control and ensuring technological initiatives align with broader organizational goals.
- **Leveraging The Full Potential of Generative AI for IT Architecture¹:** You can responsibly leverage Generative AI as a powerful augmentation tool to enhance efficiency, data-driven insights, and collaboration, provided you proactively manage its inherent risks and maintain critical human oversight

While I elaborately discuss many faces of the operating model, I want to emphasize that any Operating Model is only relevant if there are **healthy Lightweight Architectural Analytics and Collaborative Networks**. Without data and people connections, an operating model leads to an ivory tower institution, generating opinion-based decisions disconnected from reality.

Now that we've completed the high-level Grounded Architecture tour let's examine the specifics.

¹[gen-ai-potential](#)

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 examples](#)¹ and [Sokrates](#)².

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

Everywhere I worked on creating an architecture practice, I strongly (aka obsessively) **emphasized data**. Consequently, one of the first steps I make in any an architecture practice is to create an Lightweight Architectural Analytics to get a complete, up-to-date picture of critical elements of an organization's technology landscapes (Figure 1). Manual documentation does not scale, and relying on data ensures the reliability and scalability of decision-making.

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

²<https://sokrates.dev>

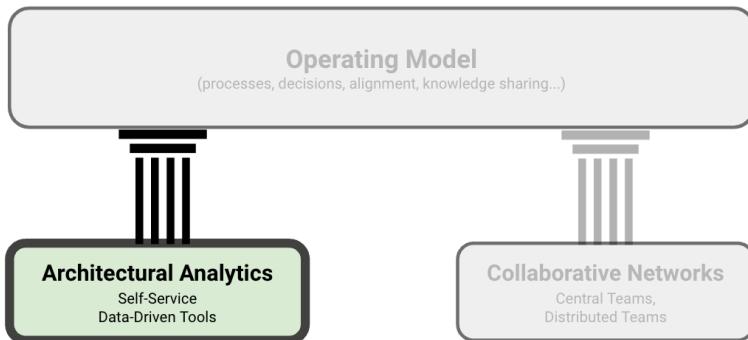


Figure 1: The Grounded Architecture framework: Lightweight Architectural Analytics.

The good news is that **big organizations have lots of data** that, if used wisely, can provide an excellent basis for an architectural Lightweight Architectural Analytics. With some **automation** and lots of **curation**, getting a crystal clear overview of the technology landscape may be closer than it initially appears.

I call the tools I will discuss in this section **lightweight** to emphasize that, you don't need to begin with costly or overly complicated software platforms. Plenty of simple and affordable tools—many of them open-source—can help you gain meaningful insights without breaking the bank.

5.1: Examples of Lightweight Architectural Analytics Tools

To illustrate what I mean by Lightweight Architectural Analytics, I will give a few concrete examples from my recent work (Figure 2).

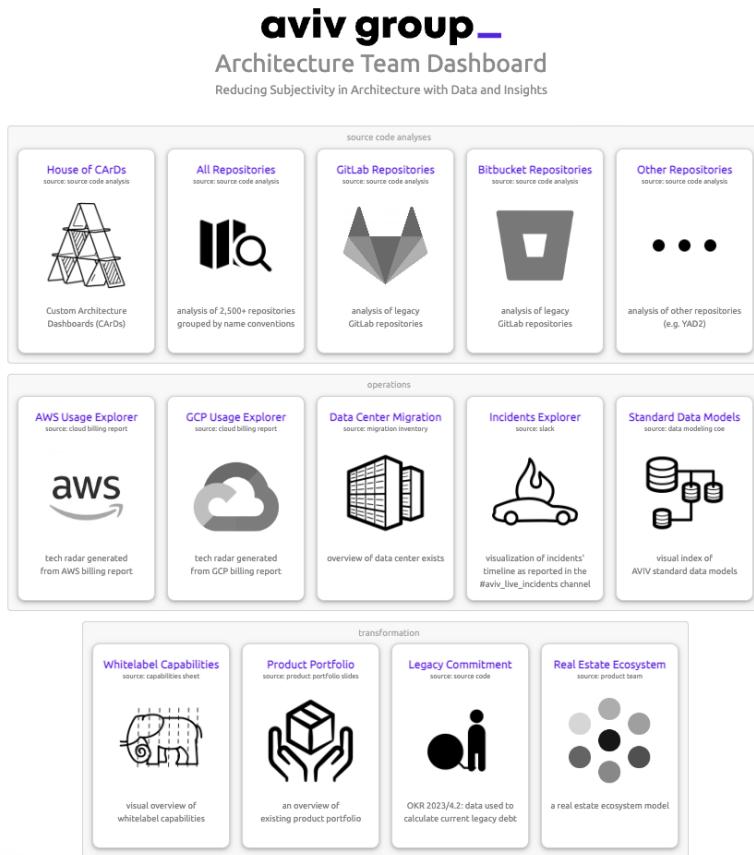


Figure 2: A screenshot of the start page of the architecture data dashboard we've built and used at AVIV Group.

I typically implement Lightweight Architectural Analytics as a dashboard

with many data apps, typically leveraging the following data sources (Figure 3):

- **Source code** contains an incredible amount of information about technology, people's activity, team dependencies, and the quality of software systems. By analyzing commit histories, code complexity, and contributions, you can identify critical areas of improvement, understand team dynamics, and ensure code quality.
- **Public cloud billing reports** provide an overview of trends in used cloud services, regions, and budgets. Monitoring billing reports can help manage budgets, identify cost-saving opportunities, and understand usage patterns across different services and regions.
- **Incident reports** can reveal trends and dependencies among incidents. Analyzing these reports can reveal trends, common issues, and dependencies among incidents, helping manage problems and improving system reliability.
- **Key business metrics**, like vibrancy, can show user activity on our systems. Tracking these metrics can help assess the business's health, understand user behavior, and guide strategic decisions to enhance user experience.
- **Activity reports from messaging and collaboration platforms (such as Slack)** can help understand discussion topics and team interactions. Analyzing these reports can help understand collaboration patterns, identify key discussion areas, and improve team communication and productivity.

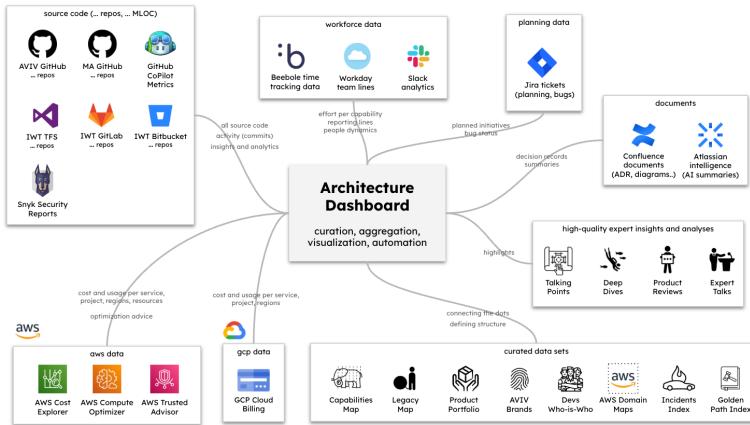


Figure 3: A overview of data sources for the architecture dashboard in the AVIV Group.

In the following sections, I detail several of these architectural data-driven tools.

5.1.1: Example 1: Source Code and Commit History Analytics

The **source code** and its **commit history** are like a treasure chest for creating data-driven architecture documentation—packed with nuggets of wisdom about **technology**, **team activities**, **dependencies**, and software quality.

Many tools can help you reveal these insights. In this section, I will focus on a simple tool I built to get these insights. To help you dig up this treasure without getting your hands too dirty, I've developed and actively maintained a free, open-source project called **Sokrates**³. Sokrates generates reports with essential source code insights (see Figure 4). Its reports are designed to be user-friendly, making it easy for a broader audience to navigate and understand their source code and its history. Sokrates is designed with an **architect's x-ray vision**, allowing you to **zoom in and out** of source code landscapes. It provides a **high-level overview** of the IT landscape, summarizing data from various teams and

³<https://sokrates.dev>

groups while also letting you dive deep into the **code-level details**. This dual functionality makes it the perfect sidekick for **CTO-level** strategy powwows and **developer-level** code critiques.

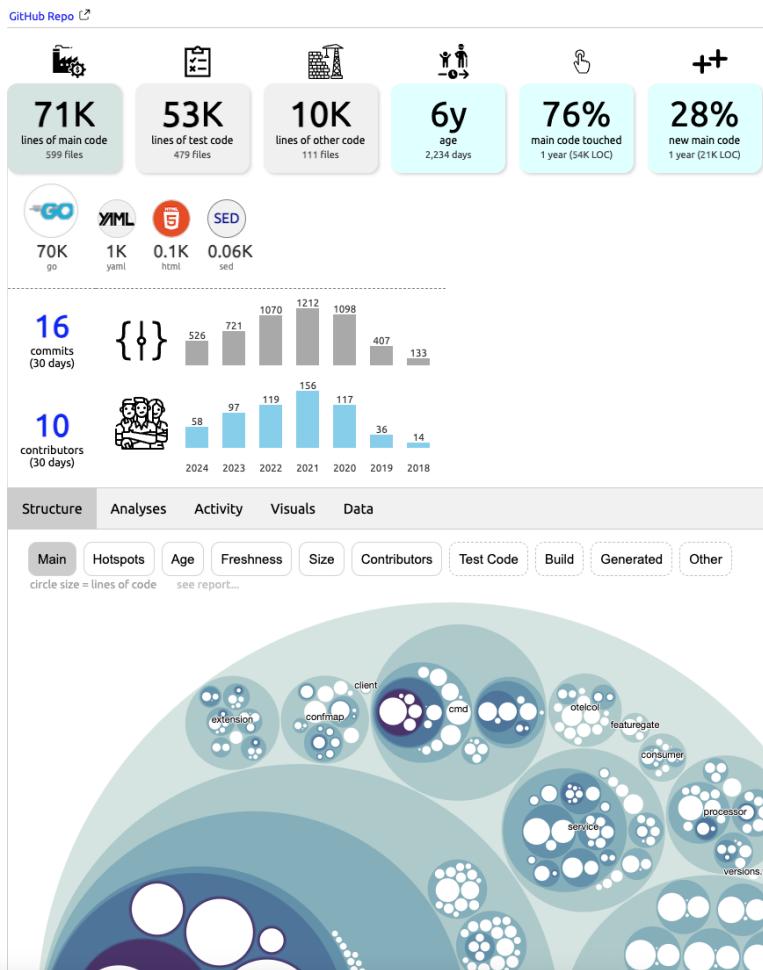


Figure 4: Screenshot of a Sokrates report.

Show me, don't tell me, you said? All right, for a more entertaining look at what Sokrates can do, check out the Sokrates examples. Here are some

blockbusters:

- **Apache Software Foundation Repositories⁴:** An epic saga of over 1,000 repositories with more than 180 million lines of code, 22,000 contributors, and 2.4 million commits.
- **Facebook/Meta OSS Repositories⁵:** A thriller with 800 repositories, 120 million lines of code, 20,000 contributors, and over 2 million commits.
- **Microsoft OSS Repositories⁶:** A drama featuring over 2,400 repositories with more than 100 million lines of code, 18,000 contributors, and 1.2 million commits.
- **Google OSS Repositories⁷:** A blockbuster with over 1,600 repositories, more than 200 million lines of code, 27,000 contributors, and 2.4 million commits.
- **Linux Source Code⁸:** A classic with 178 repository sub-folders, more than 23 million lines of code, 17,000 contributors, and 1.7 million commits.
- **Amazon OSS Repositories⁹:** A thriller with over 2,700 repositories, more than 130 million lines of code, 13,000 contributors, and 600,000 commits.

In addition to standard source code and commit history analyses, I also have built several special source code analyses to get further details:

- **Travis and Jenkins Analyzers:** Perfect for sleuthing how teams build CI/CD pipelines.
- **Dockerfile Scan:** Creates a tech radar of runtime technologies.
- **GitHub API Pull Request Analyses:** To identify deployment frequency.

Feel free to use these or similar tools, but I encourage you to experiment with your source-code analyses as well.

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

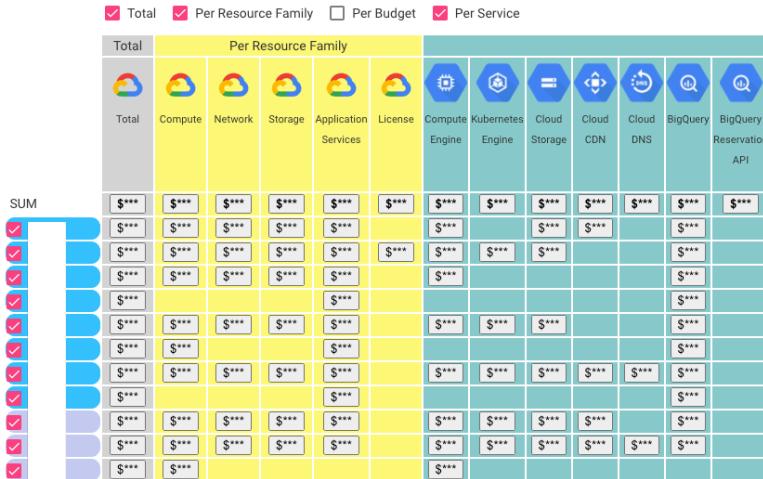
⁹https://d3axxy9bcycpv7.cloudfront.net/amzn/_sokrates_landscape/index.html

5.1.2: Example 2: Public Cloud Usage Analytics

Thanks to **uniform automation and monitoring**, using the public cloud can dramatically increase transparency. The public cloud transparency offers incredibly valuable data out-of-the-box.

[Amazon Web Services \(AWS\)](#)¹⁰, [Google Cloud Platform \(GCP\)](#)¹¹, [Microsoft Azure](#)¹², and other public cloud providers give detailed data about which platform uses which services, resource family, and budget. You can also understand which people and teams have access to each service. Getting real-time information about cloud usage and automatically understanding the trends is straightforward.

Figure 5 shows an anonymized screenshot of the Cloud Usage Explorer, a tool I built to visualize automatically collected data from standard Google Cloud Platform (GCP) usage reports.



The screenshot displays a data visualization tool for Google Cloud Platform usage. At the top, there are four checkboxes with labels: 'Total' (checked), 'Per Resource Family' (checked), 'Per Budget' (unchecked), and 'Per Service' (checked). Below this is a header row containing icons and labels for different Google Cloud services: Total, Compute, Network, Storage, Application Services, License, Compute Engine, Kubernetes Engine, Cloud Storage, Cloud CDN, Cloud DNS, BigQuery, BigQuery Reservation API. The main area is a grid where each row represents a resource family and each column represents a service. The first column, 'SUM', contains a vertical stack of colored circles (blue, pink, blue, pink, blue, pink, blue, pink, blue, pink, blue, pink, blue, pink) corresponding to the rows. The grid cells contain either a dollar sign (\$) followed by three asterisks (***), indicating a large value, or a yellow background with a black border, suggesting a specific highlight or status. The entire interface has a clean, modern design with a light blue background and white text.

Figure 5: An example of a cloud usage explorer.

¹⁰<https://aws.amazon.com>

¹¹<https://cloud.google.com/>

¹²<https://azure.microsoft.com/>

5.1.3: Example 3: Financial and Vibrancy Analytics

Finance departments are like Sherlock Holmes in the business world—super data-driven and always on the case with high-quality data that could be a goldmine for architects. Beyond the usual suspects of costs, budgets, and other dry financial stuff, I've discovered they also track the fun stuff, like vibrancy and usage levels. This data is not just interesting; it's invaluable for architects' work.

These finance operatives need this juicy data to, for instance, link the performance of their financial systems with how much they're being used. This kind of usage data is a secret weapon for architecture discussions. For instance, by connecting systems' usage levels and vibrancy with their public cloud costs, we can uncover hidden areas of improvement and inefficiencies (Figure 6).

So, next time you're knee-deep in architectural plans, don't forget to call the finance for top-notch data insights!

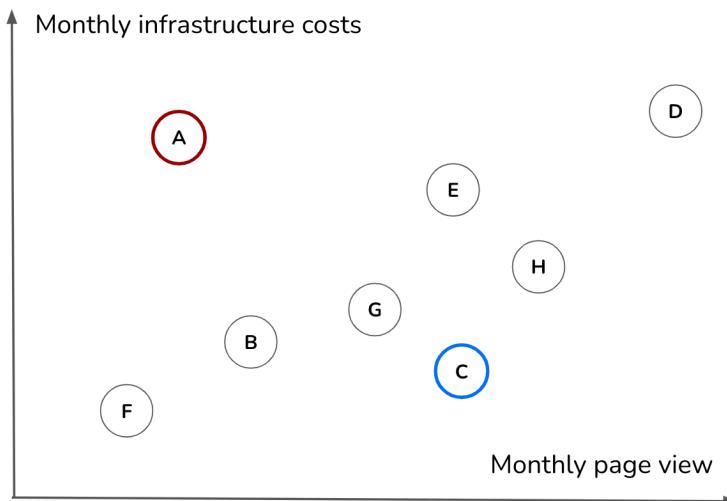


Figure 6: Combining data from different sources (e.g., cloud billing reports and vibrancy or revenue can lead to new insights (e.g., identifying inefficiencies in the application portfolio).

Nice — that's a great addition and feels totally aligned with the whole “**data as a map**” metaphor from the main text. You’re basically saying: *once you have that map, you can start asking your digital guide (ChatGPT, Gemini, etc.) for directions, insights, or even a little code to fix potholes.*

Here’s a draft for your new subsection under “**Examples of Lightweight Architectural Analytics Tools**”:

5.1.4: Example 4: Leveraging Generative AI

Once you’ve built a solid foundation of **curated, up-to-date, and accessible data**, a whole new layer of value becomes available through **generative AI tools** such as ChatGPT, Gemini, or open-source LLMs. These tools can act as intelligent companions for architectural work, transforming your data into **interactive conversations**.

Instead of static dashboards or one-off reports, you can now ask natural-language questions like:

- “What teams are committing most to legacy systems?”
- “Which services have the highest cost-to-vibrancy ratio?”
- “Can you cluster microservices based on team ownership and code activity?”
- “What are some anomalies in our incident frequency over the past 90 days?”

This approach works *only* when data is already available, curated, and reasonably structured. That’s why **Lightweight Architectural Analytics** is such a critical enabler—it ensures that the **fuel for AI analysis** is already there.

Some practical applications of this integration include:

- **Additional Source Code Insights:** Upload commit logs or Sokrates reports and prompt AI with questions about team dynamics, code complexity, or deployment patterns.
- **Exploring Cloud Usage and Cost Optimization:** Share public cloud billing exports and ask the AI to identify trends, inefficiencies, or underutilized services.

- **Incident Pattern Analysis:** Feed in incident ticket logs and ask for root cause patterns, recurring services, or teams with frequent incident ownership.
- **Business Metrics Exploration:** Ask the AI to find correlations between system vibrancy and cloud costs, or between system uptime and team size.
- **Communication Trends:** Analyze Slack activity logs to find dominant discussion themes or cross-team communication patterns.

In [the appendix¹³](#), you'll find **example prompts** and **sample curated datasets** you can upload or use as inspiration to get started.

Generative AI doesn't replace the need for good architectural judgment—but it **amplifies it**, by giving you fast feedback, exploratory analysis, and sometimes even fresh angles you might not have considered.

Interestingly, prompts themselves are more than just one-time queries—they can evolve into a **formal, reusable description of typical architectural tasks**. By writing and refining prompts, architects essentially codify their thought process and decision-making patterns.

Prompts become:

- Lightweight, transparent documentation of analysis goals
- A repeatable way to “ask the same question” of new data
- A way to share architectural thinking across teams and roles
- A medium to teach junior team members how to approach common problems

You can think of prompts as *scripts for architectural sensemaking*.

¹³[gen-ai-prompts](#)

5.2: Requirements For A Lightweight Architectural Analytics

Lightweight Architectural Analytics should be a central place with **authoritative, relevant, and curated data** about the organizational technology landscape. Technically, you can implement Lightweight Architectural Analytics tools like those discussed in the previous section, using simple resources like Google Drive, with documents organized in folders or as an internal website. I recommend investing some effort in creating better infrastructure and user experience, as it can enable more people to access and benefit from data. A solid setup will make it easier for more people to access and benefit from the data, turning it into a real asset rather than a digital junk drawer.

Simply collecting and putting data in one place will not create any value. Regardless of how you implement your Lightweight Architectural Analytics, with papers on the wall, in Google Drive, in Confluence, or with a nicely designed internal website, I have identified the following requirements that Lightweight Architectural Analytics needs to have:

- **It is the single point of truth** for all relevant architectural data. People should be able to go to one place and get relevant data.
- **It is curated for quality** so people can trust the data. Simply dumping data into one place will not help. You need to own curation to ensure that data are correct. You also should provide links to data sources so people can verify the facts.
- **It is curated for usability** so people stay focused on valuable details. You must filter out useless or less relevant details, focusing on the essence. Investing in the UX design of documents or tools you create helps.
- **It is kept up to date**, ideally in an automated fashion (or semi-automated, repeatable way).
- **It is accessible to the whole organization**. I genuinely believe that when you give employees access to information generally reserved for specialists, architects, or “higher levels,” they get more done independently. They can work faster without stopping to ask for information and approval. And they make better decisions without needing input from architects or the top.

- **It is used in decision-making.** Having nicely curated and valuable data has zero value if you cannot ensure that such data informs vital decisions.
- **It is built like a map.** Maps are some of the most crucial documents in human history—they help us store and exchange knowledge about space and place. One thing all maps do is provide readers with a **sense of orientation**. And that, in a nutshell, is what Lightweight Architectural Analytics should offer people in your organization: a sense of orientation in a waste space of technology, organizational, and business topics. The map metaphor is also helpful, as maps come with **multiple layers**. Similarly, the architecture of Lightweight Architectural Analytics should give readers data layers about systems that describe their sizes, connections, quality, security, or human activity. It's like having a trusty map that shows you where the treasure is and warns you about the dragons.

5.3: Building Lightweight Architectural Analytics

While each organization has its own quirky set of data, here are some tips I've found helpful in forming the Lightweight Architectural Analytics:

- **Start with the source code.** My motto is "*Talk is expensive. Show me the code.*" Because let's face it, code never lies—people, on the other hand, might forget a detail or two. I scan source code as soon as possible using tools like [Sokrates](#)¹⁴. Modern IT enterprises store almost everything as code. It's the richest and most up-to-date documentation on what's happening. Quick source code scans can reveal that your "simple" system is actually a digital spaghetti monster.
- **Connect with finance and governance teams.** My second motto is "*Follow the money!*" You'd be amazed at what you can learn from finance data (minus the sensitive parts, like revenue projections—let's keep those secrets safe). Cloud billing reports and tech usage trends are collected anyway. Extract and connect these to get new insights without pestering people for more details.
- **Maintain a culture of transparency.** A culture of transparency means having an organizational environment where open communication, honest information sharing, and accountability are actively encouraged and practiced at all levels. Such a culture allows simplicity, as you need less complex authorization mechanisms.
- **Own the curation.** People need to trust your data like they trust their morning coffee to wake them up. Spend time understanding data sets, curate them, and ensure they're consistently presented. Think of yourself as the master curator and chief UX designer of Lightweight Architectural Analytics.
- **Use simple and easy-to-maintain infrastructures.** For example, I publish the results of Sokrates analyses and other data tools as static resources on our enterprise GitHub pages. Avoid the headache of complex databases and backend software. In the [Architecture Dashboard Examples repository](#)¹⁵, you'll find the source code for building the Architecture Data Dashboard. The dashboard is a

¹⁴<https://sokrates.dev>

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

simple static website generated from JSON files and published via GitHub pages¹⁶.

These tips might save you from drowning in the sea of data chaos and make your architectural life a bit smoother—or at least give you a few laughs.

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

5.4: Using Lightweight Architectural Analytics

Lightweight Architectural Analytics can churn out data by the bucketful. Think of it like an atlas or a map—it's great for finding your bearings and understanding the lay of the land. But, with the **right mindset**, you can turn that data into a treasure trove of insights.



image by cofotoisme from istock

Interpreting and using data requires a bit of effort—think of it as a **detective game** where the data holds the answers, but you need to come armed with the right questions. Here are some of the questions you should ask when you've got a pile of data at your fingertips:

- **Are we all rowing in the same direction?** Source code overviews, public cloud usage explorers, or tech radars can highlight when systems and teams are out of sync, sparking heated debates that lead to real action.
- **Are we making the most of our technology?** Comparing usage trends between teams can reveal fascinating outliers—both the virtuosos and those who are... let's say, still tuning their guitars.
- **Are there signs our systems might need a little TLC (tender, loving care)?** Look out for oversized systems, rampant duplication,

and files that go on longer than your Aunt Marge’s vacation slideshows.

- **Productivity trends: is more really more, or is more actually less?** For instance, comparing the number of git merges to the number of developers can reveal if our dev processes are scalable. When scaling up teams, we aim to speed up delivery, but without proper structure, we might end up with a digital mosh pit.
- **Are we collaborating the way we want to?** Repository analysis can reveal team topologies and unwanted dependencies. Sometimes, teams collaborate like a well-oiled machine; other times, it’s more like a group project in high school.
- **Are we working on what we really want to?** We may aspire to innovate, but if we’re spending most of our time wrestling with legacy maintenance, we might need to rethink our priorities.

So there you have it. The data’s ready to spill its secrets—you need to know the right questions to ask. So, what is your question?

5.5: 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>

5.6: 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 gaps between business, product, organizational, and technology issues. They're like the Swiss Army knives of the tech world. Hiring architects is like searching for a unicorn who's also a full-stack developer with a knack for diplomacy. They need not only in-depth technical knowledge but also domain-specific and organizational knowledge.

So, you cannot just 3D print your architects or hire them in buckets. But you can carefully organize, empower, and leverage this scarce talent (Figure 1).

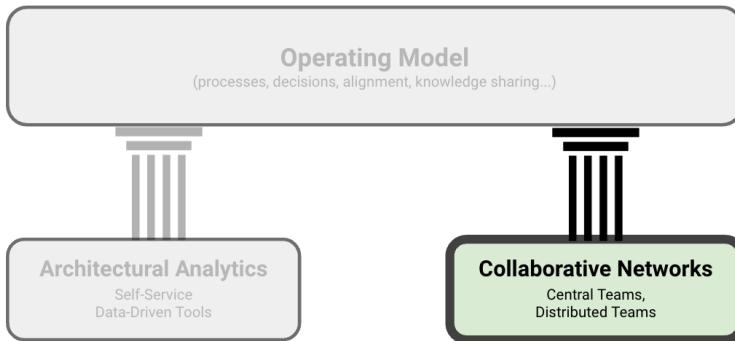


Figure 1: The Grounded Architecture framework: Collaborative Networks.

A strong network of people doing architecture is crucial for any real

impact. In simpler terms, **Strong Architecture = Strong Architects**.

In my recent escapades, I worked with two teams of architect teams: a **small central architecture team** and a **cross-organizational distributed virtual team**. The central team is like the wise elders, guiding and supporting the rest of the organization. The distributed virtual team, on the other hand, is a merry band of rebels, working locally but also connecting across the organization, increasing transparency, building networks, and implementing change. Together, such a mix of central and distributed teams can create a robust **collaborative network** driving change across an organization.

6.1: Background: Central vs. Federated Architecture Practice

An IT architecture practice generally follows one of two fundamental models: central or federated (Figure 2, [McKinsey 2022¹](#)).

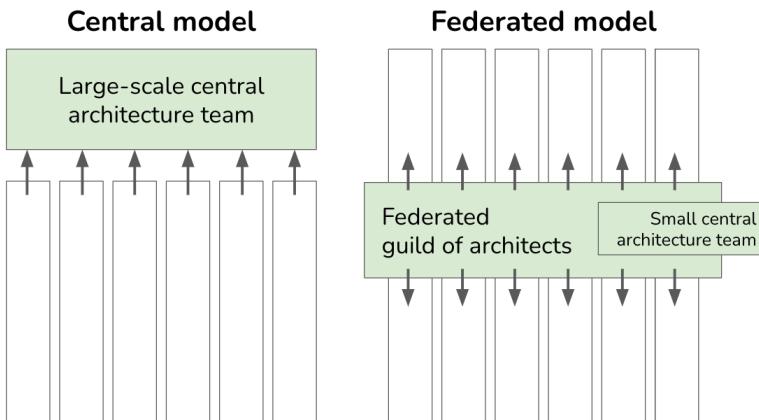


Figure 2: Central vs. Federated Architecture Practice.

The **central model** is like a tightly run coffee shop. A large central team sets the rules, approves new work, and ensures everyone follows the script. Development teams in this model are like customers with no coffee-making skills, relying entirely on the central baristas. The central team handles infrastructure, operations, and security—essentially controlling the caffeine supply chain.

The **federated model** is more like a coffee co-op, where each team has its own barista. A small central team or an **architecture center of excellence (CoE)** might provide high-level guidance, but the real magic happens locally. These baristas (architects) are embedded in development teams, facilitating high-level planning and acting as on-demand service providers.

¹<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/tech-forward/crafting-the-optimal-model-for-the-it-architecture-organization>

The **federated model** is a favorite among **cross-functional DevOps cultures**. It integrates infrastructure, operations, and security into its brews. Architects in this model focus on **facilitation and enablement**, and stay close to the action.

6.2: The Hybrid Model

To operate in a complex context, you must invest effort to ensure you have the **right people at the right places**. In the end, I usually found it best to adopt a model of a hybrid organization combining elements of central and federated orientation structures:

- A Small Central Architectural Team, and
- Architecture Guilds & Virtual Architectural Teams.

This model is similar to the previously described **federated model** but with extra investment in a central team that should be more than just an on-demand service provider.

Think of it as a symphony orchestra, where the central team is the conductor, and the guilds and virtual teams are the talented musicians playing different instruments. Sure, each musician can play their part solo, but without the conductor, it might sound more like a chaotic jam session at your neighbor's garage.

I prefer the **hybrid team structure** as it scales better in complex organizations:

- Guilds and virtual architecture teams support execution by **increasing the number of people involved** in architectural activities and increasing work efficiency through better alignment. Members representing various organizational units can have much **more impact across the board**.
- Having some **capacity on the central level** serves as a **catalyst**, helping people at local levels to do their jobs while being aligned and better connected with overall strategic goals and other teams working on related topics.

6.2.1: Central Architecture Team

The roles of people in central teams may differ depending on the organization. However, it's crucial to recognize the unique value a central architecture team.



image by simonkr from istock

Such a team, in addition to the doing architecture work, can be instrumental in covering these often overlooked responsibilities:

- **Build and maintain the architecture [Lightweight Architectural Analytics](#).** With all AI advances, it will not build itself! You need clear ownership, curation, and technical support to make it work.
- **Promote data-informed decision-making.** It's not enough to have data; you've got to *use* it. This is like trying to convince your grandpa to use a smartphone—challenging but doable. Architects should be tech-savvy wizards who show everyone else how things are done, making data the critical actor of every decision.
- **Proactively identify, connect, and maintain relationships with all relevant stakeholders.** Think of architects as the social butterflies at a high school reunion—they've got to bridge all the cliques. Connecting different organizational units and stakeholders is their superpower.
- **Build internal architecture communities and guilds.** Organizing rituals and gatherings takes effort.

While guilds and virtual teams can handle many of these activities, their voluntary and sometimes laid-back nature makes their support less reliable. The central architecture team can step in like the dependable backup

generator, taking full long-term ownership and ensuring continuity, even if the community vibe fizzles out.

6.2.2: Architecture Guilds & Virtual Architecture Teams

I've always found it crucial to rally passionate troops about architecture through a guild, a community of interest, or a virtual team. After all, who doesn't love a good architecture geek meet-up?



image by sdi productions from istock

Our guilds or virtual team members typically double as architects or tech leads in their respective departments. But they also moonlight by collaborating with their counterparts from other areas. These **peer-to-peer communities** are collectively responsible for spotting and nurturing architectural talent, mentoring, and helping each other out of sticky situations.

If you end up having an overabundance of guilds and teams, you can try splitting them into different cliques:

- **General or core teams:** These folks could tackle various general architectural topics.
- **Specialist teams:** These teams can focus on specific parts of the tech stack, such as native mobile apps, web frontends, or public clouds.
- **Strategic initiatives teams:** These teams work on the big picture, such as data strategy, public cloud strategy, transactions, or verticalization.

Connecting the central and distributed teams is essential. In most places I've worked, we organized gatherings to facilitate regular alignments, such as:

- **Regular (e.g., bi-weekly) forums:** Here, updates are shared, architectural spikes are announced, and decisions are debated.
- **Annual or bi-annual summits:** These are the architecture equivalent of Comic-Con, with several days of intense knowledge sharing and workshops.
- **Ad-hoc workshops:** These focus on specific topics and deep dives into niche subjects.

While the central team can provide essential support, all communities must take the initiative and get many people involved in these events. It's crucial that people are **active participants** rather than passive spectators to ensure more engagement and commitment. So, get ready to roll up your sleeves and dive in, because architecture is a team sport, and everyone needs to play their part!

6.3: Building Collaborative Networks

While each organization will need its unique approach, here are some tips I found helpful when forming architecture teams and building Collaborative Networks:

- Before making grandiose plans for reorganizations, **connect with the people already doing architectural work** in an organization. Bring together all the critical tech leaders, regardless of their position and title. Being well-connected to these folks will be crucial in any architecture organization, so this effort is never wasted.
- If creating a virtual team is a part of your architecture strategy, move away from the informal community of practice and start **building a team with more accountability and responsibility**.
- **Connect with non-architecture people** early to gain broader support for building architecture teams and guilds. Being well-connected to these stakeholders is crucial to have strong collaborative networks.
- **Avoid hiring a digital hitman**². Instead, invest in growing internal talent. Think of it this way: architects need to know the technology, the domain, and the organization.
- **Externalize**. Reach out and connect. Participate in external events. Publish your work. Being strong externally can help you grow and attract architectural talent. Everyone wants to join the band when you're rocking the stage.

²<https://architectelevator.com/transformation/dont-hire-hitman/>



image by chantellev from pixabay

6.4: To Probe Further

- Agile and Architecture: Friend, not Foe³, by Gregor Hohpe, 2020
- Crafting the optimal model for the IT architecture organization⁴, by Christian Lilley et al., 2022
- Developers mentoring other developers: practices I've seen work well⁵, by Gergely Orosz, 2022

³https://architectelevator.com/transformation/agile_architecture/

⁴<https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/tech-forward/crafting-the-optimal-model-for-the-it-architecture-organization>

⁵<https://blog.pragmaticengineer.com/developers-mentoring-other-developers/>

6.5: Questions to Consider

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

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

7: Operating Model: General Principles



image by ma_rish from istock

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

KEY POINTS:

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

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

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

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

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

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

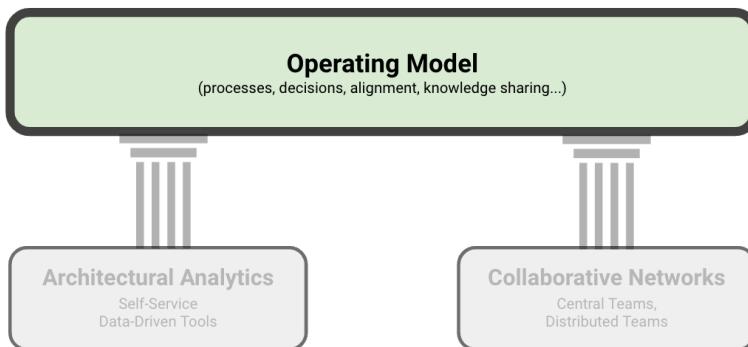


Figure 1: The Grounded Architecture framework: Operating Model.

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

7.1: Examples of Architecture Activities

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



image by brauns from istock

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

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

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

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

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

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

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

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

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

7.2.1: High-Level Operating Framework

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

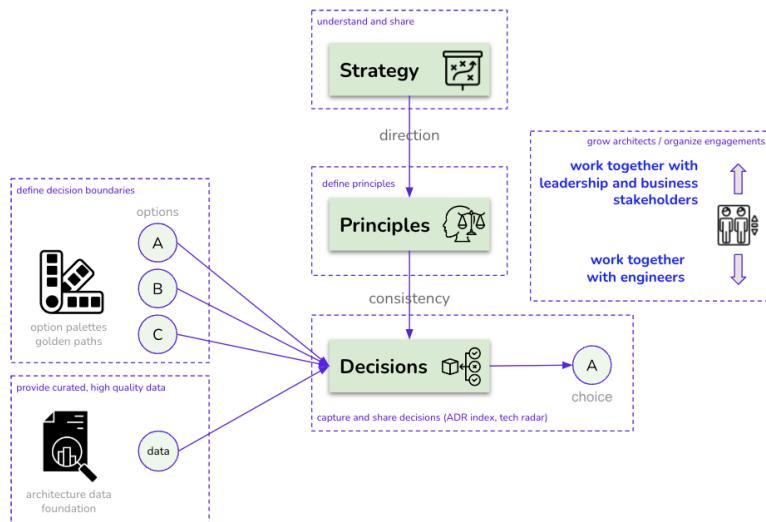


Figure 2: A common operating framework I typically use for Grounded Architecture activities.

Here are the key characteristics of this operating framework:

Engagement mindset:

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

Contributions of architects:

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

Social dynamics of architects:

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

Shift left:

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

7.2.2: Distributing Decisions, Autonomy, and Alignment

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

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

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

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

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



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

7.2.3: General Architecture Decision Policy

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

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

“Decide in the Organization’s Best Interests.”

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

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

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

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

7.2.4: Golden Paths

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

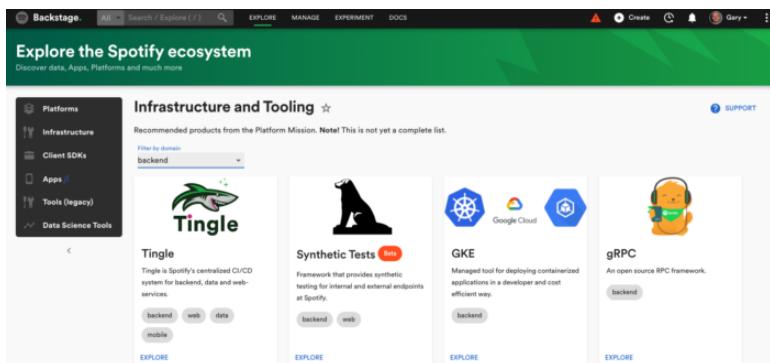


image by engineering.atspotify.com

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

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

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

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

7.3: Embracing Diversity

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

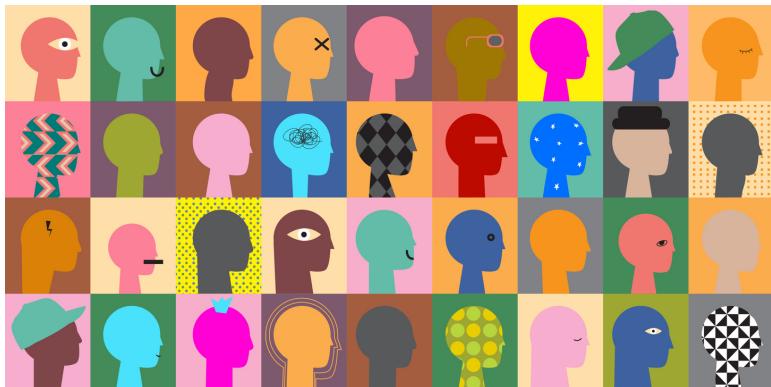


image by annaspoka from istock

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

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

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

7.4: Setting Boundaries

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



image by ingenui from istock

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

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

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

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

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

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

7.5: To Probe Further

- [Scaling the Practice of Architecture, Conversationally⁴](https://martinfowler.com/articles/scaling-architecture-conversationally.html), by Andrew Harmel-Law, 2021
- [Scaling Engineering Teams via RFCs: Writing Things Down⁵](https://blog.pragmaticengineer.com/scaling-engineering-teams-via-writing-things-down-rfcs/), by Gergely Orosz, 2022
- [Transformation Agents: An Engagement Model⁶](https://architectelevator.com/transformation/transformation-engagement-model/), by Gregor Hohpe, 2022
- [Would you like architects with your architecture?⁷](https://architectelevator.com/architecture/organizing-architecture/), 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-rfcs/>

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

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

7.6: Questions to Consider

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

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

8: Cooperation-Based Operating Model: Six Simple Rules



image by nanostockk from istock

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

KEY POINTS:

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

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

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

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

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

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

8.1: Background: Hard and Soft Management

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

8.1.1: Hard Approach

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

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

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



image by cyano66 from istock

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

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

8.1.2: Soft Approach

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

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



image by alessandro biascioli from istock

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

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

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

8.2: Collaboration Approach

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

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

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

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

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

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

8.3: Rule 1: Understand What Your People Do

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



image by scythers5 from istock

8.3.1: Guidelines for Understanding Performance

To genuinely understand performance, consider the following principles:

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

8.3.2: Leveraging Architecture Practice

Architecture practice can significantly aid in understanding organizational behaviors through:

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

8.4: Rule 2: Reinforce Integrators

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



image by robert owen_wahl from pixabay

8.4.1: Guidelines for Reinforcing Integrators

To strengthen integrators within your organization, consider these strategies:

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

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

8.4.2: Enhancing Integrators Through Architecture Practice

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

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

8.5: Rule 3: Increase the Total Quantity of Power

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



image by prostock_studio from istock

8.5.1: Guidelines for Increasing Power Quantity

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

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

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

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

8.5.2: Enhancing Power Quantity Through Architecture Practice

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

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

8.6: Rule 4: Increase Reciprocity

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



image by natnan srisuwant from istock

8.6.1: Guidelines for Enhancing Reciprocity

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

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

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

8.6.2: Enhancing Reciprocity Through Architecture Practice

Architecture practice can significantly increase reciprocity within an organization:

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

8.7: Rule 5: Extend the Shadow of the Future

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



image by joe from pixabay

8.7.1: Guidelines for Extending the Shadow of the Future

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

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

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

8.7.2: Extending the Shadow of the Future Through Architecture Practice

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

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

8.8: Rule 6: Reward Those Who Cooperate

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



image by stocksnap from pixabay

8.8.1: Guidelines for Rewarding Cooperation

To effectively reward those who cooperate, managers should:

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

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

8.8.2: Rewarding Cooperation Through Architecture Practice

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

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

8.9: To Probe Further

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

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

8.10: Questions to Consider

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

9: Operating Model: 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 framework of rules, practices, and processes by which an organization is directed and controlled. It encompasses the mechanisms by which an organization's goals are set, pursued, and monitored, ensuring accountability, fairness, and transparency. Governance can be applied to various domains, including corporate, IT, project, and data governance.

IT architecture is a form of governance because it **establishes structured frameworks** for managing and controlling an organization's technology resources and processes. It ensures alignment with business objectives, promotes standardization, manages risks, optimizes resources, facilitates change management, supports decision-making, measures performance, and fosters innovation. IT architecture governance involves defining the technological infrastructure, setting standards for technology use, and ensuring that all technology-related activities align with the organization's overall goals. This form of governance provides a comprehensive approach to managing technology, ensuring that it supports the organization's strategic direction and operational needs.

The difficulty of governance stems from the need to navigate a complex web of diverse interests, rapidly changing conditions, and multifaceted

challenges. There is no one-size-fits-all form of governance, as each organization has unique needs, goals, and environments. Effective governance requires adaptability, collaboration, and a commitment to addressing immediate and long-term issues. It involves balancing various stakeholder interests, anticipating and responding to external changes, and continually refining governance practices to meet evolving demands. This complexity demands a strategic approach, robust communication, and a willingness to innovate and adapt to ensure that governance frameworks remain relevant and effective in achieving organizational objectives.

Architecture practice should support governance models that are aligned and adaptable to organizations' complex and diverse needs. Consequently, I see an architecture governance model as a well-balanced hybrid of three different styles of governing:

- **nudging,**
- **taxes (economic incentives), and**
- **mandates and bans.**

9.1: Nudging

In behavioral economics and psychology, a **nudge** is a **subtle or indirect suggestion** influencing someone's behavior or decision-making **without forcing them or limiting their freedom of choice**. Nudges can be applied in various settings, such as policy-making, marketing, and personal interactions, to encourage people to make better choices, improve their well-being, or achieve specific goals.



image by liudmila chernetska from istock

A nudge can take many forms, such as a slight change in the environment, a gentle reminder, positive reinforcement, or a default option. For example, placing healthy food options at eye level in a cafeteria can nudge people to choose healthier meals. Setting a default option for organ donation can increase the number of donors.

The concept of a nudge was popularized by the book "**Nudge: Improving Decisions About Health, Wealth, and Happiness**" by Richard Thaler and Cass Sunstein, which argues that various cognitive biases and heuristics often influence people's decisions, and that nudges can help people overcome these biases and **make better choices**.

Richard Thaler and Cass Sunstein also introduced the concept of choice architecture as a critical component of nudging. Choice architecture refers to how options are presented to individuals, which can significantly influence their choices. It is the design of the decision-making environment, which includes the layout, structure, and organization of available options.

In IT architecture, examples of nudging include:

- Architectural **principles** as informal decision guidelines. Such principles do not prescribe a solution but can subtly guide alignment.
- Recommendations for **best practices** to stimulate introduction and alignment around such practices,
- Default options for technology choices via **golden paths**¹
- **Highlighting** bad quality software on a **Lightweight Architectural Analytics** dashboard to create subtle pressure for people to improve it,
- Tracking of **tech debt** to create awareness about its size and lead action to reduce it,
- **Visualizing cost trends** of cloud services per team to stimulate teams to improve the performance efficiency of their software.

Nudges can frequently lead to better alignment and more harmonization without the negative consequences of mandates, bans, or taxation.

Grounded Architecture is well aligned with ideas of nudging. I designed many **Lightweight Architectural Analytics** tools to **highlight areas and issues** we wanted (nudged) people to improve. The **Collaborative Networks** can create mechanisms for sharing experiences, promoting **positive examples**, and capturing lessons learned to help people make better, more informed decisions. In the **Operating Model**, I use the operating model that stimulates people to make decisions autonomously but **nudges them to stay well-aligned** and connected to the organizational strategic direction.

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

9.2: Taxation (Economic Incentives)

Governing with taxes is a strategic approach where individuals or departments are not prohibited from making choices or decisions. Instead, they are required to “pay” some form of tax on the resources they use. This system creates a feedback loop that encourages responsible resource consumption and helps optimize overall system efficiency. One practical application of this approach is in managing the costs of public cloud usage within an organization. By cross-charging these costs across various departments or projects, each unit receives a clear signal about their resource consumption. This not only helps in allocating costs accurately but also promotes awareness and encourages efforts to minimize unnecessary usage, effectively preventing the “tragedy of the commons,” where unrestricted access to shared resources can lead to overconsumption and depletion.



image by steve buissinne from pixabay

Compared to nudging, which influences behavior subtly by providing information or setting default choices without imposing direct consequences, taxes introduce tangible consequences. For example, projects that exceed budgeted IT costs due to excessive resource consumption might be canceled. The role of an architecture practice in this form of

governance is crucial. It involves ensuring that taxation policies are based on accurate and comprehensive data, using public cloud cost reports and other relevant information to inform tax rates and policies. Transparency is essential, allowing all stakeholders to understand how and why taxes are levied and providing clear insights and reports detailing the basis of taxation and its impact on resource consumption.

Developing efficient feedback loops is another critical aspect, providing timely and actionable feedback on critical metrics related to taxes and continually refining and optimizing the taxation system. [Lightweight Architectural Analytics](#) plays a key role in this approach by aggregating and analyzing all data related to resource consumption and taxation, generating insights from public cloud cost reports, and guiding decision-making. Collaborative Networks ensures alignment of organizational processes, goals, and working methods with the taxation system, fostering a culture of responsible resource usage and continuous improvement.

In conclusion, governing with taxes is a robust approach to resource management that balances autonomy with accountability. By implementing a data-driven and transparent taxation system, organizations can optimize resource usage, prevent overconsumption, and drive meaningful change. An architecture practice, supported by strong Lightweight Architectural Analytics and Collaborative Networks, is essential in achieving these goals and ensuring the sustainability of shared resources.

9.3: Mandates and Bans

Governing with mandates and bans involves guiding people by explicitly defining what they should or should not do. In various workplaces, such mandates and bans have played a limited yet important role in defining the broader strategic boundaries of choices available to people. For instance, restricting the use of public cloud providers to specific vendors or adhering to strict privacy and security procedures needs to be explicitly defined and controlled.

Using bans with care and as a last resort is essential to avoid unnecessary blocking or slowing down development and innovation. However, mandates and bans can help clarify critical topics where nudging or taxation would not be sufficient. For example, having clear rules and control mechanisms to avoid breaking privacy or financial laws can prevent unnecessary incidents and damage. Explicitly defined mandates and bans can ensure compliance with important regulations and safeguard the organization's integrity and reputation.



image by tumisu from pixabay

The role of architecture in this form of governance should be to act as a **stakeholder but not the sole owner in defining mandates and bans**. These mandates and bans should often be determined collaboratively with other functions, such as security and legal departments. The an architecture practice can contribute by **creating clarity and providing transparency**.

[Lightweight Architectural Analytics](#) is crucial for creating **clarity and transparency**. For example, it can provide insights through security reports or maps of areas in the source code or infrastructure that need monitoring and controlling based on privacy or security requirements. This foundation helps in identifying and mitigating risks, ensuring that the organization's technology landscape aligns with its governance frameworks.

The [Collaborative Networks](#) can help propagate the decision and ensure its **positive impact and acceptance**. Mandates and bans should not be issued routinely without sufficient explanation. It is crucial to spend time with all stakeholders to explain the **reasons and motivations** behind introducing certain limitations. A strong Collaborative Networks fosters strong connections with key stakeholders, leveraging these relationships to implement changes smoothly. This foundation ensures that governance measures are understood, accepted, and integrated into the organization's culture and practices, enhancing overall effectiveness.

In summary, governing with mandates and bans involves setting clear, explicit guidelines for acceptable behavior and practices. It requires a balanced approach to ensure it does not stifle innovation while maintaining necessary controls. Effective implementation relies on collaboration, transparency, and communication to ensure that all stakeholders understand and accept the governance measures.

9.4: Questions to Consider

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

10: Transforming Organizations with Grounded Architecture



image by gremlin from istock

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

KEY POINTS:

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

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

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

10.1: Executing at Scale

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

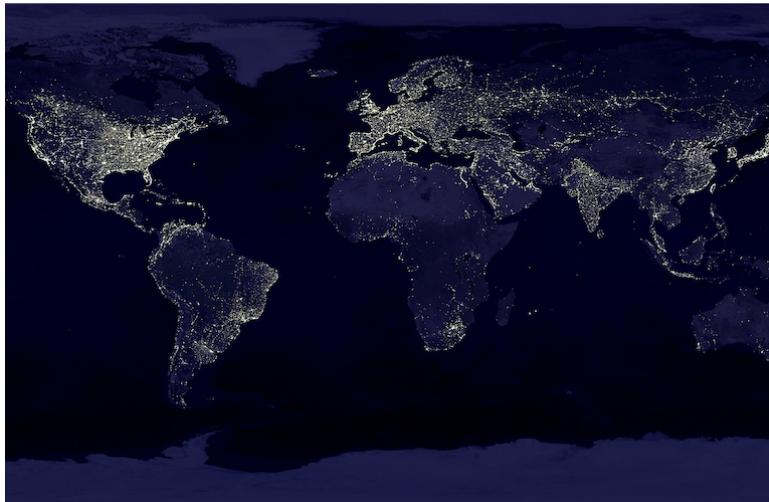


image by wikiimages from pixabay

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

10.1.1: Lightweight Architectural Analytics

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

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

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

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

10.1.2: Collaborative Networks

The Collaborative Networks enhances execution at scale by focusing on:

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

10.1.3: Operating Model

The Operating Model promotes execution at scale by:

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

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

10.2: Adaptivity

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

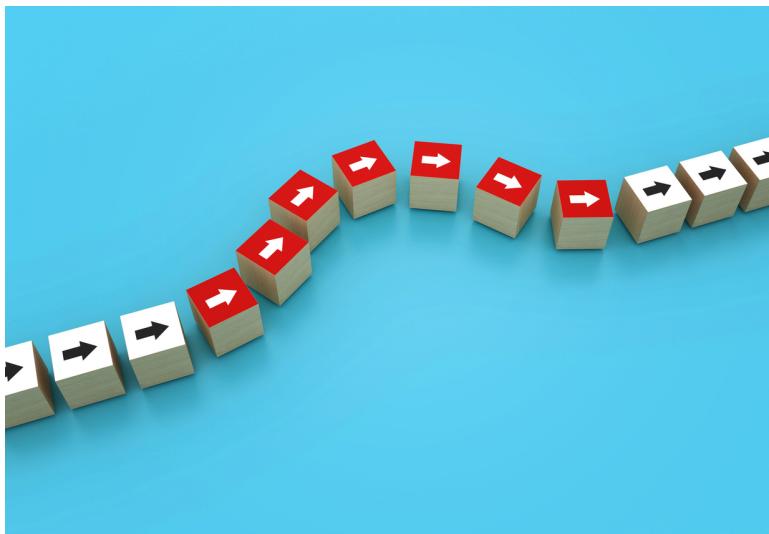


image by porcorex from istock

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

10.2.1: Lightweight Architectural Analytics

The Lightweight Architectural Analytics can support adaptability by:

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

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

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

10.2.2: Collaborative Networks

The Collaborative Networks can support adaptability by:

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

10.2.3: Operating Model

The Operating Model can support adaptability by:

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

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

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

10.3: Improving the Quality of Decision-Making with Data

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



image by cofotoisme from istock

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

10.3.1: Lightweight Architectural Analytics

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

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

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

10.3.2: Collaborative Networks

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

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

10.3.3: Operating Model

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

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

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

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

10.4: Maximizing Organizational Alignment

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



image by 36clicks from istock

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

10.4.1: Lightweight Architectural Analytics

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

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

10.4.2: Collaborative Networks

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

10.4.3: Operating Model

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

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

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

10.5: Maximizing Organizational Learning

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

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



image by rawpixel from istock

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

10.5.1: Lightweight Architectural Analytics

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

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

10.5.2: Collaborative Networks

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

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

10.5.3: Operating Model

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

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

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

10.6: Questions to Consider

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

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

11: Next Steps: Leveraging The Potential of Generative AI

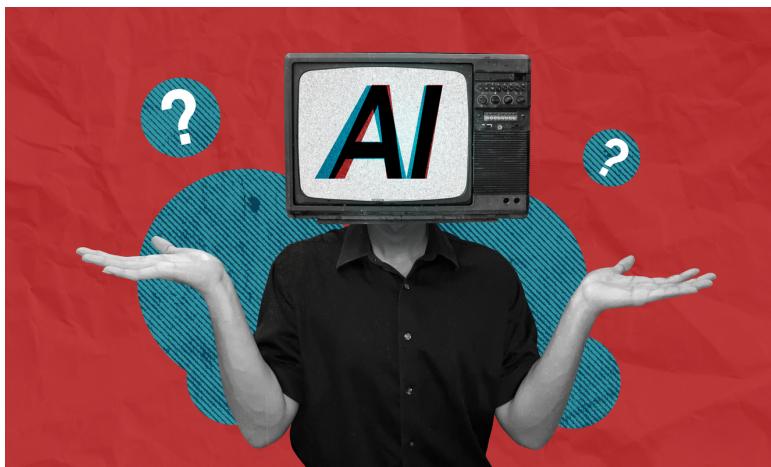


image by mininixx doodle from istock

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

KEY POINTS:

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

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

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

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

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

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

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

¹[gen-ai-prompts](#)

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

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



image by lemon tm from istock

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

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

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

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

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

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

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

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

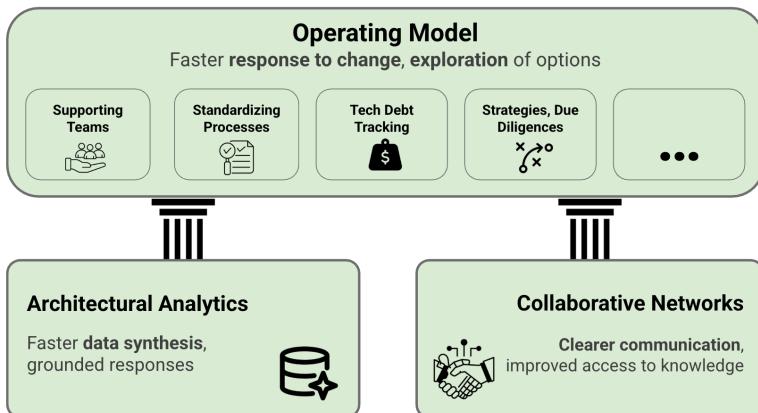


Figure 1: An overview of potential Generative AI potential in the context of the Grounded Architecture Framework.

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

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

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

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

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

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

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

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

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

11.3: Practical Examples: GenAI in Action

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



image by blackjack3d from istock

11.3.1: Use Case 1: Analysing Dependencies and Technical Debt

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

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

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

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

11.3.3: Use Case 3: Creating Architecture Diagrams from Descriptions

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

11.3.4: Use Case 4: Improving Requirements Analysis

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

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

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

11.3.6: Use Case 6: Enhancing Code Review Processes

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

assess if it adds to technical debt.

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

11.3.7: Use Case 7: Drafting Communications for Stakeholders

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

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

11.4: Balancing Benefits, Risks, and Challenges

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

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

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

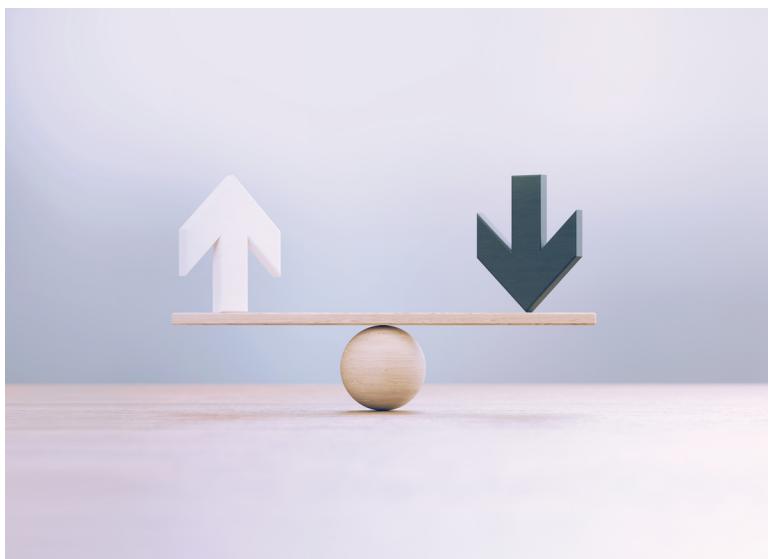


image by microstockhub from istock

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

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

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

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

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

11.5: Best Practices for Responsibly Adopting GenAI

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



image by aree sarak from istock

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

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

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

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

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

11.6: The Evolving Landscape

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



image by galeanu mihai from istock

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

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

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

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

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

11.7: Questions to Consider

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

11.8: Tables

11.8.1: Grounded Architecture Element vs. GenAI Capability

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

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

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

11.8.2: GenAI Checklist

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

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

11.9: To Probe Further: References

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

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

11.9.2: Retrieval-Augmented Generation (RAG)

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

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

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

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

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

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

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

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

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

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

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

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

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

11.9.3: Agentic AI

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.9.4: Responsible AI, Governance, Best Practices & Security

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.9.5: GenAI & Enterprise Architecture / Solution Architecture

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11.9.6: GenAI Use Cases & Tools (Specific Areas)

Cloud Operations:

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

Code Review & Development:

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

Documentation & Diagramming:

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

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

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

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

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

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

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

⁶⁸<https://www.coderrabbit.ai/>

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

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

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

Requirements Analysis:

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

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

⁷²<https://swapp.ai/>

⁷³<https://www.klarity.ai/architect>

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

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

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

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

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

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

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

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

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

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

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

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

Technical Debt Management:

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

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

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

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

⁸⁸<https://arxiv.org/pdf/2306.10194>

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

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

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