

[Open in app](#)

Search



The Different Types of Software Architects : C4 model perspective



Pier-Jean Malandrino

Published in Scub-Lab

8 min read · Jan 5, 2024

 Listen Share More

This paper proposes a description of different architecture types. However, as this has been done many times before, I want to add the perspective of the C4 model to help understand who will intervene and at each level and with whom they will have to interact.

Software architecture is a critical aspect of software development. It involves the high-level structuring of software systems to meet technical and business requirements. Software architects play a pivotal role in this process by making design choices, dictating technical standards and leading implementation efforts. There are various types of software architects, each specializing in different aspects of software systems. This paper analyzes the standard types of software architects, highlighting their roles, responsibilities, and impact on software development processes.

Types of Software Architects

Enterprise Architect

The Enterprise Architect ensures that the organization's technological infrastructure aligns with its business strategy. This role integrates the IT strategy with business goals and governs compliance with company policies and regulations. Their role involves overseeing the integration of various IT components to ensure they function cohesively in support of organizational objectives.

Broad Vision: Focuses on aligning IT strategy with business goals.

Governance: Ensures that all aspects of the technological environment adhere to the company's policies and regulations.

Integration: Oversees the integration of various IT aspects to ensure they work harmoniously towards the organizational objectives.

Key Differences with the CTO : I have been asked many times what is the difference between the Enterprise Architect and the CTO. The key differences between these roles lie in their organizational hierarchy, focus, responsibilities, and approaches. The CTO, as a top-level executive, directs the broader business strategy and technological vision, engaging in strategic decision-making, innovation, and external advocacy for technology. In contrast, the Enterprise Architect, typically occupying a senior-level management position within the IT department,

concentrates more internally on the design, governance, and optimization of IT infrastructure, ensuring its alignment with business processes. While the CTO adopts a strategic view, focusing on technology in the context of business growth, the Enterprise Architect takes a more tactical stance, dealing with the specifics of IT infrastructure and its operational effectiveness. Both roles are vital for an organization's technological success, with the CTO shaping the overarching technology direction and the Enterprise Architect focusing on the practical design and efficiency of IT systems in support of the company's strategy.

The Enterprise Architect could be seen as the technical right hand of the CTO.

Solution Architect

The Solution Architect acts as a link between business challenges and technological solutions. They design and lead the implementation of solution architecture across projects or programs, providing essential technical guidance and coaching to developers and engineers. Additionally, they manage project scope, ensuring that solutions align precisely with specific business requirements.

Solution Development: Designs and leads the implementation of a solution architecture across a project or program.

Technical Guidance: Provides technical guidance and coaching to developers and engineers.

Project Scope Management: Ensures that the solutions meet the specific business needs within the defined scope.

Technical Architects

Technical Architects are experts who possess in-depth knowledge in a specific *domain* and specialize in specific areas of expertise, often functioning within the broader framework of the Enterprise Architecture Team or contributing to various delivery projects. The term “domain” in this context refers to a niche area of knowledge, encompassing a range of specialized skill sets. These architects play critical roles in different aspects of software architecture, each focusing on a unique domain:

Application Architect: Concentrates on the design and structure of individual applications, ensuring they meet both technical and business requirements.

Technical Architect: Deals with the technical infrastructure and hardware aspects, ensuring that the technology infrastructure supports the specific requirements of the domain.

Security Architect: Focuses on designing robust security structures, ensuring that the domain's architecture is safeguarded against potential threats and vulnerabilities.

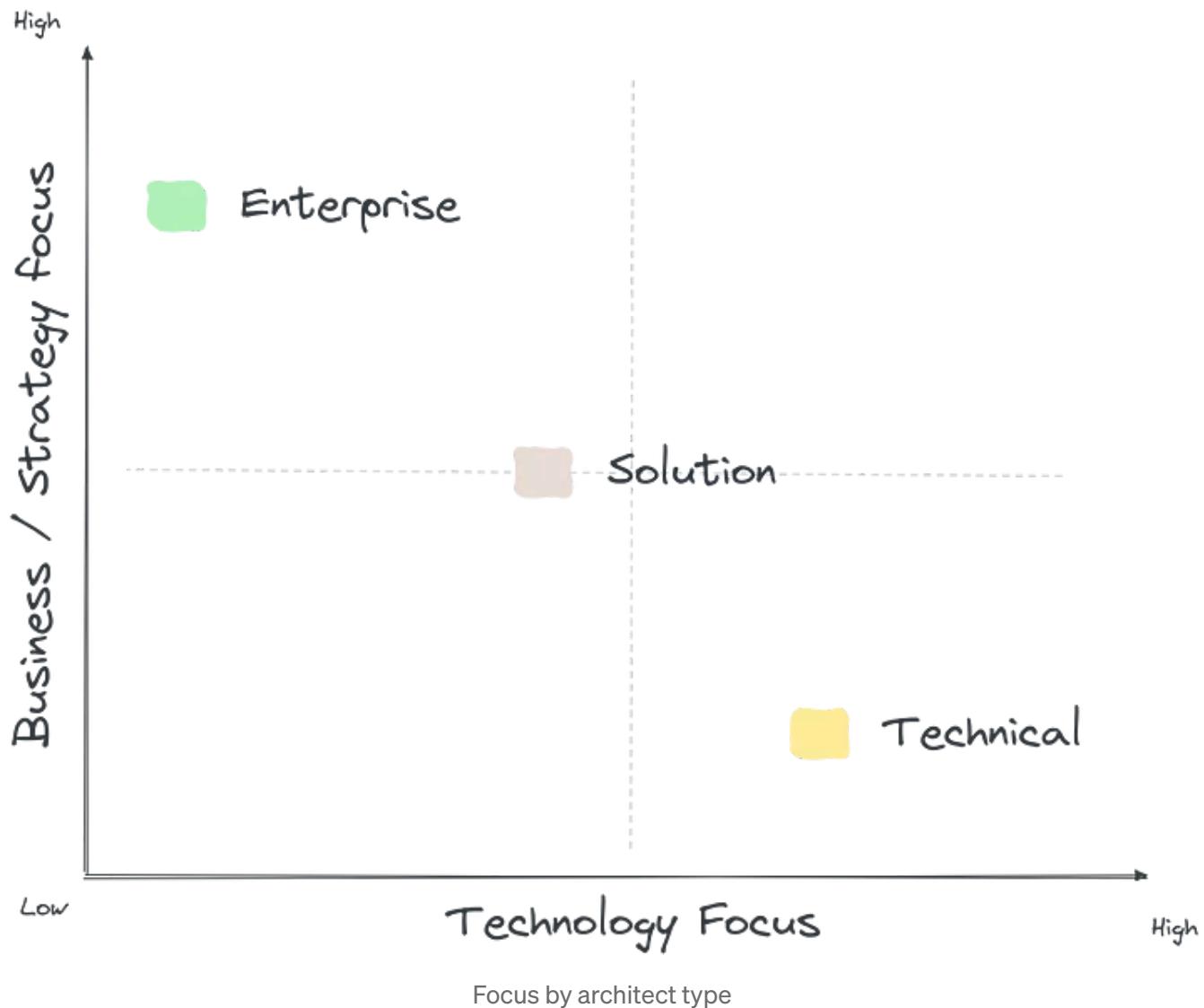
Business Architect: Focuses on aligning business strategy with technological solutions, ensuring that business processes are optimally supported by technology.

Data Architect : Plays a crucial role in ensuring effective leverage of data assets to support the organization's decision-making processes. They develop and manage the data strategy, policies, standards, and practices, design data models and structures to support business operations, and ensure data accuracy and integrity across systems.

Cloud Architect : They are a key in facilitating an organization's transition to cloud computing, optimizing cloud solutions for performance, cost, and scalability. This role involves designing cloud architecture strategies, developing cloud solutions, overseeing the migration of systems to cloud platforms, and managing relationships with cloud service providers.

Conclusion

In essence, the field of software architecture is defined by three primary types of architects: Enterprise Architects, Solution Architects, and a diverse group of Technical Architects. Each type focuses on different aspects of software development, with some emphasizing strategy and others delving into technological details. Enterprise Architects align the organization's technology with business goals, whereas Solution Architects bridge business needs with technical solutions. Technical Architects, encompassing roles like Application, Data, and Security Architects, specialize in various domains, providing depth in specific technical areas. Together, these architects create a comprehensive approach to software architecture, ensuring both strategic alignment and technical excellence across multiple fields.



The Business Architect is not focused on technology but rather on the business domain . They are a particular case.

Parallel with the C4 model

The C4 Model for software architecture provides a framework for visualizing and documenting the software architecture of a system at different levels of abstraction. It consists of four hierarchical levels: Context, Containers, Components, and Code. Each level targets a specific set of concerns, and different types of architects can play key roles at each of these levels.

You can read my paper on the C4 model here if you are not familiar with C4 model :

Architecture Modeling : C4 Model

What is the C4 Model?

lab.scub.net

Let's parallel this with the roles of various architects and see where they might intervene.

Context

Focus: The highest level showing how the system in focus interacts with users and other systems.

Relevant Architect: Chief Technology Officer (CTO) and Enterprise Architect.

Role: The CTO ensures that the system aligns with the broader business and technological strategy. The Enterprise Architect focuses on how the system fits within the wider IT landscape and its external interactions.

Containers

Focus: Zooms into the system to illustrate the high-level technology choices, showing how responsibilities are distributed across it.

Relevant Architect: Enterprise Architect and Solution Architect.

Role: The Enterprise Architect designs the high-level structure and identifies integration points with other systems.

The Solution Architect defines the technology stack and oversees the architectural decisions for each container (e.g., web applications, mobile apps, databases).

Components

Focus: Delves deeper into the containers to reveal the internal components and their interactions.

Relevant Architect: Solution Architect and Technical Architect.

Role: The Solution Architect structures the components within a container, ensuring they align with the solution's goals.

The Technical Architect details the specific technologies and patterns used, focusing on aspects like scalability, reliability, and performance.

Code

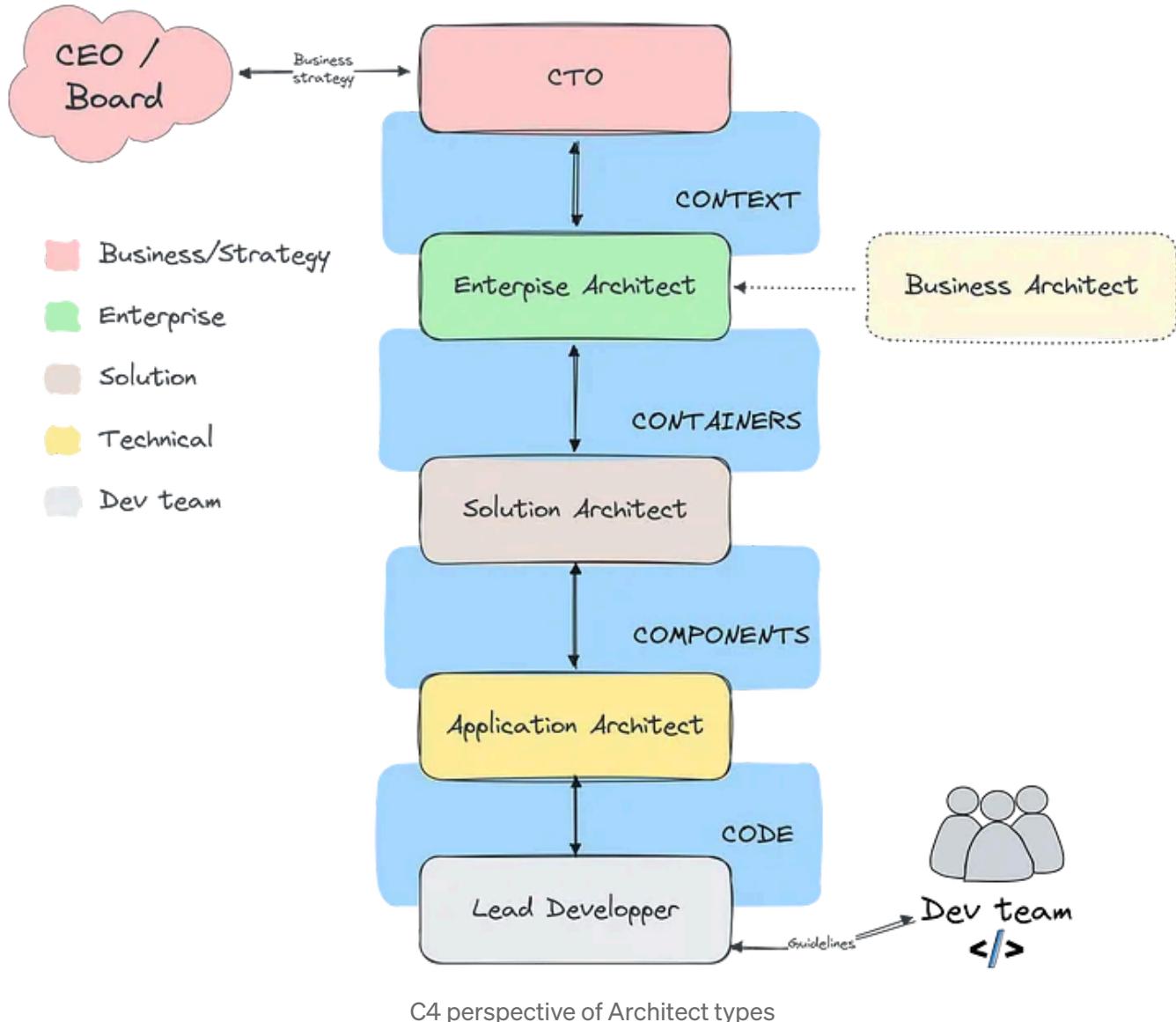
Focus: The lowest level, detailing the implementation of individual components.

Relevant Architect: Application Architect and Technical Architect.

Role: The Application Architect is involved in defining the code structure, frameworks, and coding standards.

The Technical Architect ensures that code-level decisions align with the technology strategy and standards.

Summary



CTO: Involved at the Context level, aligning the system with business strategy and technological innovation.

Enterprise Architect: Active from the Context to Containers level, focusing on system integration and alignment with enterprise IT strategy.

Solution Architect: Engages primarily at the Containers and Components level, designing the solution architecture within the system.

Application Architect: Primarily active at the Code level, ensuring best practices in software development and implementation.

Business Architect : Although he may be considered a “*Technical Architect*”, his domain skills enable him to integrate the context level and assist in decision-making for his specific business domain. He may also appear at other levels under particular conditions.

Each architect plays a particular role at different levels of the C4 model, ensuring that the architecture is robust, scalable, and aligned with both the technical and business objectives. This layered approach allows for clear separation of concerns, making complex software systems easier to understand, communicate, and manage.

Conclusion

This analysis of the different types of software architects, enriched by the perspective of the C4 Model, not only clarifies their individual roles and areas of expertise but also opens up a broader understanding of how these roles interconnect and contribute to the overall success of software development. It highlights the essential nature of collaboration and the importance of having diverse architectural expertise within a project.

Architects are seekers of solutions, which inherently means they are adept describers of problems.

Despite the varied roles and specializations of different types of software architects, there are core attributes and approaches common to all. Primarily, they must *embody pragmatism, steering clear of dogmatism* in their methodologies. This pragmatism is key in balancing ideal architectural models with the practical constraints and realities of business and technology environments. Furthermore, they are seekers of solutions, which inherently means they are adept describers of problems. Their ability to accurately identify and articulate challenges is as crucial as their skill in devising effective solutions. This dual role of problem identifier and solution provider underpins their effectiveness in driving technological advancements and ensuring the alignment of IT strategies with business objectives. Their expertise not only lies in creating robust architectures but also in foreseeing

potential pitfalls and proactively addressing them, thereby ensuring sustainable and adaptable software systems. These common traits form the backbone of their roles, making them indispensable in the ever-evolving landscape of software architecture.

I am the CTO and Head of an architectural unit in a digital company. I participate in the development of technological strategy, design solutions, and lead R&D projects.

Thank you for reading! If you enjoyed this article, please feel free to  and help others find it. Please do not hesitate to share your thoughts in the comments section below.

Scub Lab

Thank you for being a part of our community! Before you go:

- Be sure to *clap* and *follow the writer!* 
- You can find even more content at lab.scub.net 
- Sign up for our free weekly newsletter. 
- Follow us on Twitter(X), LinkedIn, and our site web.

Technology

Software Development

Software Engineering

Software Architecture

Design Patterns



Edit profile

Written by Pier-Jean Malandrino

3.5K Followers · Editor for Scub-Lab

CTO & Head of Architecture at a digital firm, I drive technological strategy, design innovative solutions, & lead R&D projects.

More from Pier-Jean Malandrino and Scub-Lab



Pier-Jean Malandrino in Scub-Lab

Architecture : The cheat sheet

This paper presents a concise summary of various software architecture patterns, methods and models.

5 min read · Dec 25, 2023



1.92K



6



...



 Boris Bodin in Scub-Lab

What is it ?—Gherkin for test naming

Learn how using Gherkin syntax for test naming improves test readability and maintenance in SpringBoot/Angular applications.

◆ · 9 min read · Nov 13, 2023

 5 



...



 Boris Bodin in Scub-Lab

Focus on: Factory Pattern

Demystifying the Builder Pattern: A practical guide for software architects and developers.

◆ · 11 min read · Dec 11, 2023

👏 6 💬

✚ ⌂ ...



 Pier-Jean Malandrino in Scub-Lab

Architecture Anti-Patterns : The DARK side of the Architect

Amid the realm of logic and structured thought, architects, much like the mythical creatures of old, harbor a shadowy underside. Like the...

3 min read · Oct 31, 2023

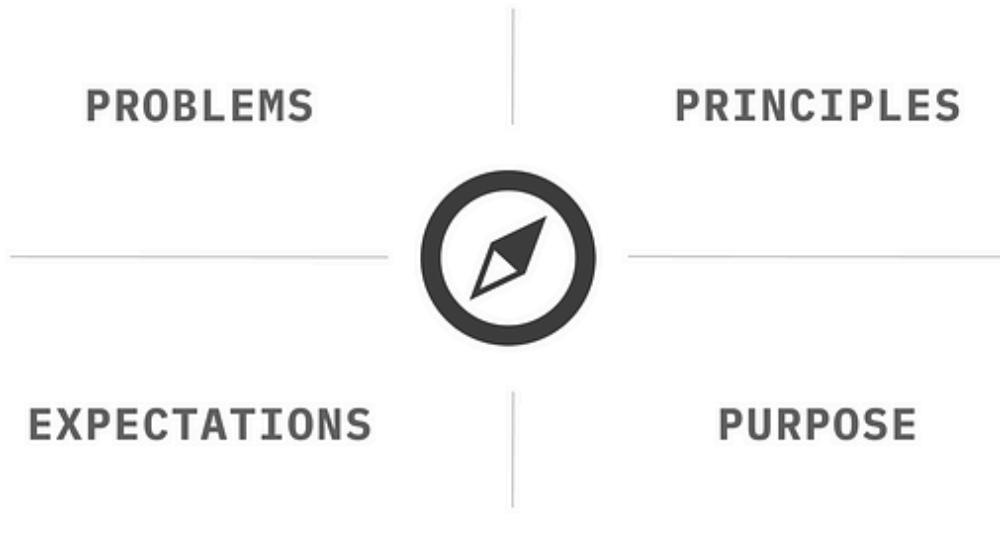
👏 434 💬 7

✚ ⌂ ...

See all from Pier-Jean Malandrino

See all from Scub-Lab

Recommended from Medium



 Practical Engineering

 Mirek Stanek in Practical Engineering Management

How to lead a team of senior engineers

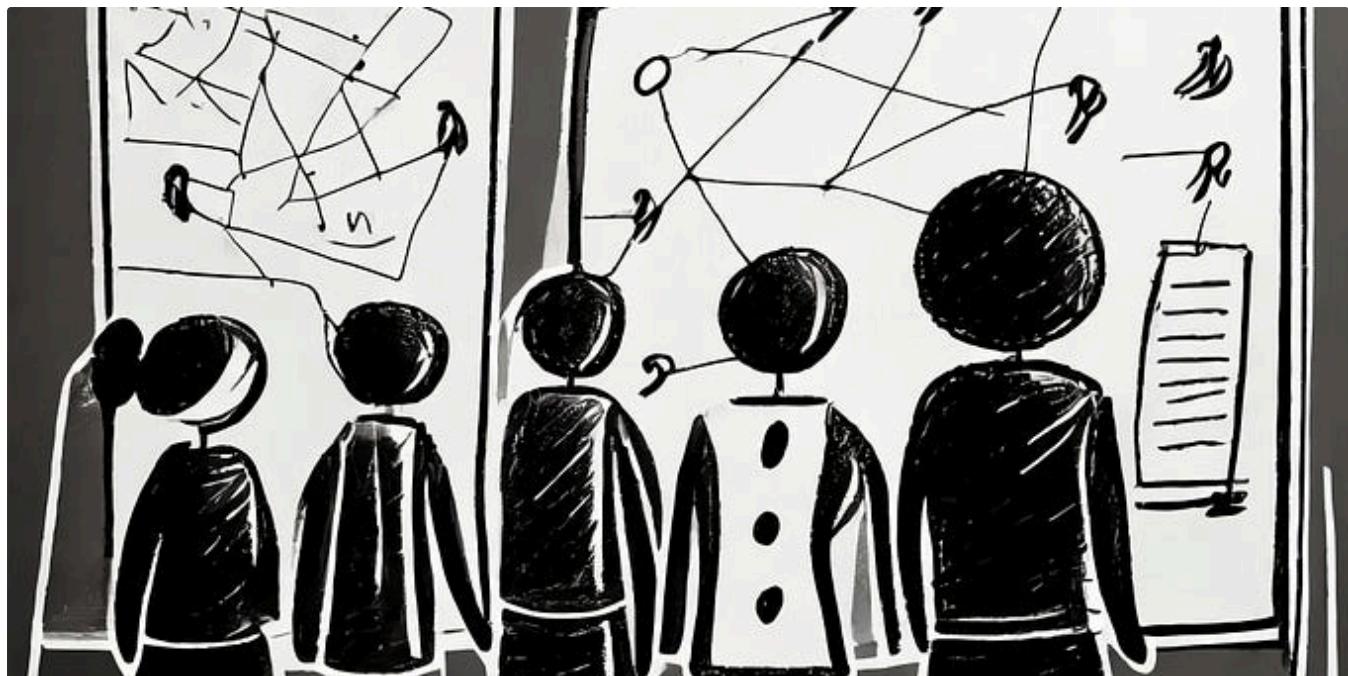
Managing people who are more skillful than you

8 min read · Jan 17, 2024

 1.6K  14



...



 Daniel Moldovan

How to be a staff engineer. Or team lead. Or principal engineer. Or manager. Or whatever ...

Many wonder what does it take to become a senior software engineer. Or staff engineer. Or principal software engineer. Or manager. Or ...

15 min read · Jan 11, 2024

 200  1



...

Lists



Stories to Help You Grow as a Software Developer

19 stories · 1053 saves



General Coding Knowledge

20 stories · 1210 saves



AI Regulation

6 stories · 448 saves



Coding & Development

11 stories · 610 saves



 Aashish Peepra

How to design clean API interfaces

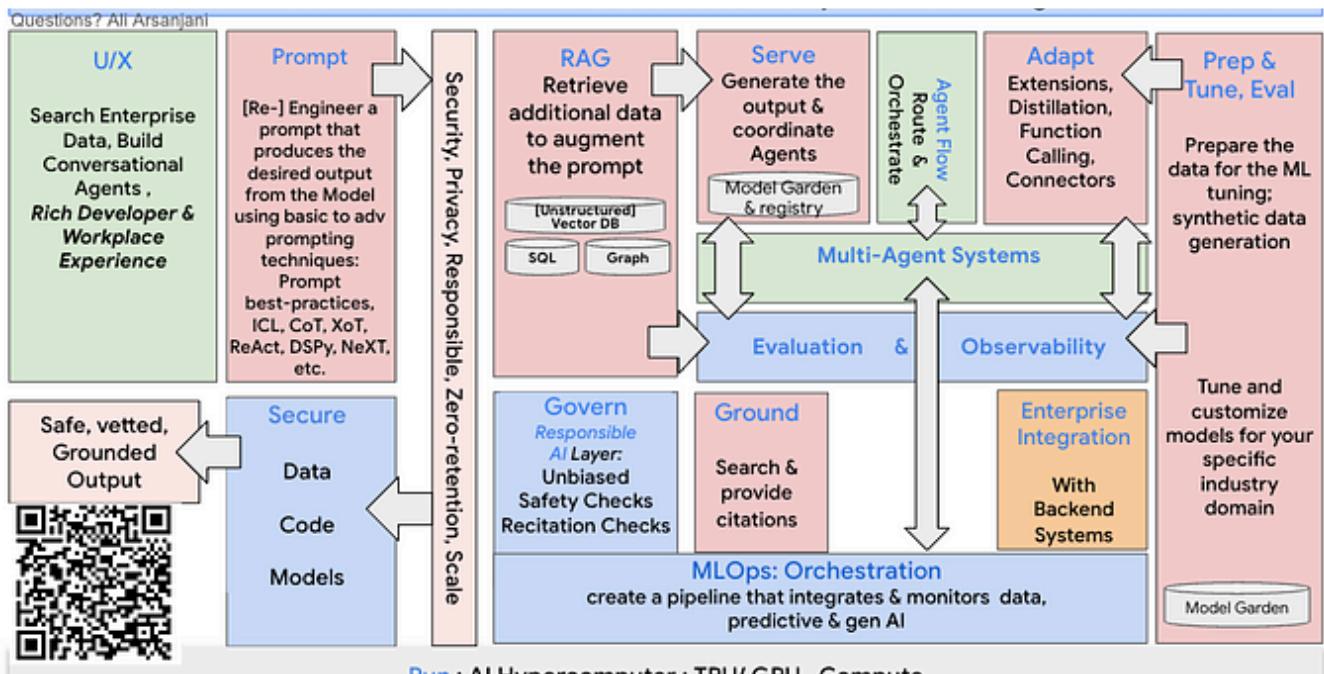
This is going to be a tough read. This contains too much code to read and no pictures. It's not for people with faint hearts. So read if...

7 min read · Mar 3, 2024

 525  7



...



 Ali Arsanjani

The GenAI Reference Architecture

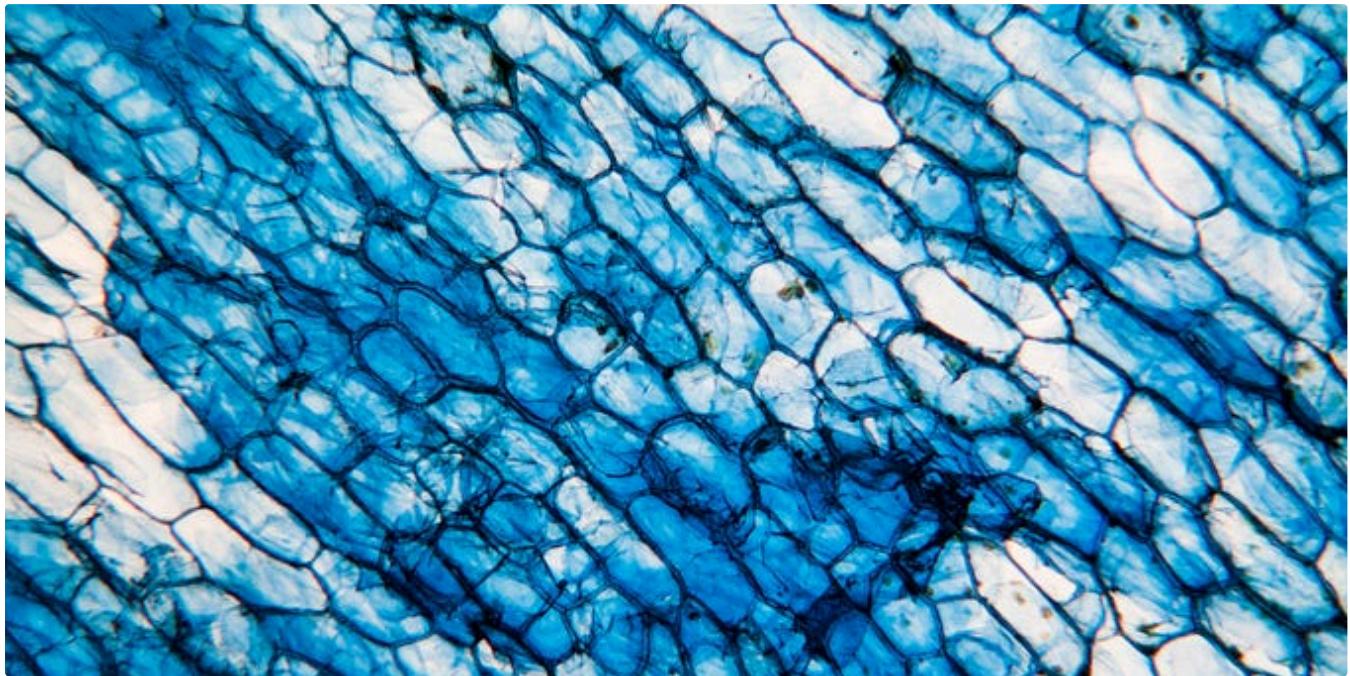
26 min read · Apr 29, 2024

614

6



...



chubernetes

Evolution of Monolithic Systems

The Entropy of Software in Tech Startups

13 min read · May 5, 2024

152

1



...

Software design principles



Pavlo Kolodka in Level Up Coding

The 20 Essential Principles of Software Development: LoD, SoC, SOLID, and Beyond.

Core software development principles that every developer must know.

19 min read · Apr 14, 2024

1.4K

16



...

See more recommendations