Cloud Architecture Demystified

Understand how to design sustainable architectures in the world of Agile, DevOps, and Cloud

Keshri Asthana

Ankur Mittal

www.bpbonline.com

First published: 2023

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55518-262

www.bpbonline.com

Forewards

> **What you leave behind is not what is engraved in stone monuments, but what is woven into the lives of others.**
>
> — *Pericles*

It's my pleasure to know Keshri and Ankur in a relationship that has grown over the decades from being colleagues, who were fortunate to work on projects from Tokyo to New York, to thought partners, benefiting me (most of the times!) with unlimited access to their deep knowledge and context in all things architecture.

A common theme across many counters we shared is the dizzying effect of the trichotomy - disorientation brought on by the evolving technology landscape, of growing awareness that systems design being potent means of differentiation and the increasing need for certitude in an otherwise uncertain world requires practitioners (across enterprise consumers and producers of technology) be constantly catching up and keeping up for want of staying ahead. By itself, this is unsustainable. From the experience, one would default to the binary options to solve this conundrum – to reduce the amount of perceived risk (by trying to play catch-up) or by increasing our tolerance for change. However, rarely do edge scenarios help solve the complexity enterprises are challenged with today.

Cloud Architecture Demystified lays out the principles across the ends by providing a continuum of options. You will find the reasoning and rationale of interest and intrigue. Hopefully, it will help you incorporate some or several of these principles in your systems journey.

— Parminder Bansil
(Our Lifelong Mentor & Guide)

Information Technology is well known for its buzzwords and hyped trends. Using Gartner's famous Hype Cycle model, I can say many new trends don't go beyond the Hype Phase. More importantly, only few adopters make the Realization Phase in reality. Among these buzzwords is The Cloud - A panacea for all timeless CTO woes. To many, moving their datacenter to an external cloud is a job well done and worth a celebration.

Real realization of value from Cloud implementations will need very intense strategization.

Architecture, design, code, the way we build, the way we test, the way we deploy, the way we troubleshoot/monitor may need to undergo an overhaul.

Cloud Architecture Demystified is a must read for all CTOs and Technology Heads involved in taking Cloud Strategy decisions related or executing Cloud adoption projects. It comprehensively covers all the elements that must be considered to extract the real value that Cloud offers. It has been written in simple and few words with loads of examples.

I was fortunate to have worked with Keshri on some exciting and yet challenging IT projects. He was an architect with utter clarity of how he wanted to build the software that delivers business value, is scalable, can survive the hard times software is subjected to, is debuggable and is designed to evolve forever. In my rather long career in IT, there are very few professionals who could do justice to this book and Keshri is one of them. I am glad that he took the initiative to capture all his learnings from real life situations that he experienced in his career into this book.

— Ravi Pratap Singh
(Our Lifelong Mentor & Guide)

About the Authors

- **Keshri Asthana** has been working in software development for more than 22 years, playing central roles in numerous projects as an architect, technical leader, and software engineer, delivering projects using Open source and Microsoft Technologies for big companies, including well succeeded projects in India, Japan, United States, United Kingdom and Singapore. Currently, he is Principal Program Manager with Microsoft. He is also an accomplished postgraduate completing a Masters in Business Administration and has many Microsoft certifications in Azure and Web Development technologies. Furthermore, the author participates as a speaker in Conferences and has successfully contributed to large projects run by multiple enterprises and the Government of India. He is one of the most respected architects in Microsoft and partner Ecosystem.

- **Ankur Mittal** has been working in software development for more than 21 years, playing central roles in numerous projects as an architect, technical leader, and software engineer, delivering projects using Microsoft Technologies for big companies, including well succeeded projects in India, Japan, and the United States. Currently, he is a Senior Cloud Solution Architect with Microsoft. He is also an accomplished postgraduate completing a degree in computer applications and has many Microsoft certifications in Azure and Web Development technologies. Furthermore, the author participates as a speaker in IT Conferences and has successfully contributed to large projects run by multiple startups and the Government of India. He is one of the most respected architects in Microsoft and partner organizations.

Acknowledgement

We would like to express our deepest gratitude to the following individuals who have supported and inspired us throughout the process of writing this book:

First and foremost, we would like to thank our families for their unwavering support and encouragement. Without them, this book would not have been possible.

We would also like to thank Parminder Iqbal Bansil and Ravi Pratap Singh for agreeing to spend some time in writing the forward for us. They have long been mentors and high impact business problem solvers. They have solved and architected various businesses and technical problems that their experience can be a topic for another book. Thank you, Sir's'!!

We would like to acknowledge our editor, BPB Publications, for their guidance, feedback, and patience throughout the editing process. Their expertise and insight have been invaluable.

We are grateful to our mentors. They have ensured we stay on our course and learn to contribute to the world using the best assets we have at our disposal. The lateral thought process that they have built in us has enabled us to think in multiple dimensions.

We would like to extend my thanks to the reviewers of this book, whose insightful comments and suggestions helped us to improve the manuscript.

Finally, we would like to thank my readers, whose interest and enthusiasm for this project has been a constant source of motivation.

Thank you all for your support and encouragement.

Preface

Ankur and I joined the Microsoft at the same time. We joined the Cloud Bandwagon that Microsoft was starting. Both of us came from very similar coding backgrounds, we both worked in project and product companies. We both worked in a wide spectrum of process environments; where we had, on one side CMMI Level 5 and on other side cooking and eating one day at a time. Although we had similar backgrounds, our approach varied with different sets of customers. Ankur worked with various teams of startups having altogether a different problem of scale and I worked mostly in enterprise where process and procedure is key and we both exchanged ideas how business priority impacts architecture and this is something we tried to jot down in this book!

When we joined Microsoft, we got an opportunity to work with lots of companies and a huge number of people. We got a chance to work in different combinations of people, process, and technology. Every interaction was a learning opportunity for us.

One harsh reality that we came across was diminishing architectural practices. We saw a lack of technology vision, lack of enterprise architecture, lack of software architecture and no one in an organization having end-to-end vision. There were several excuses, agile methodology, changing technology, Software Vendor Affinity etc.; none of them really standing test of time.

We saw technology decisions were more cost driven rather than other more important aspects. Technology and architecture decisions were being taken by people who did not have architectural vision. Technology leaders of today discount the importance of architecture and architects in their process and organization. We did see some organizations investing in architecture and reaping benefits. You will read some examples in the chapters.

The knowledge base that we gathered talking with people, understanding their process, their limitations, practical approaches etc. motivated us to write this book. It seems appropriate, however, to offer this little book, reflecting essentially our personal views.

Although we both originally grew up in the programming side of software engineering, we had some great mentors that we would like to thank them for as we feel they have contributed to this book too.

Our own conclusions are embodied in the essays that follow, which are intended for professional architects, technology leaders, and especially the agile architects.

Although written as separable essays, there is a central argument contained. We strongly believe that, irrespective of development method followed, architecture and end-to-end vision is a super critical and mandatory part of software development. We believe the critical need to be architecturally perseverant for conceptual integrity of an IT organization. Some of our chapters challenge some traditional thoughts and are sure to trigger out-of-box thinking.

For example, when humankind starts colonizing Mars, will address databases need to change? Will we need to add one more field in our database, "Planet"? Will it mean that all databases and all UI will need to change? Will it need another Y2K like effort?

Few of our chapters also explore other aspects of software engineering management.

The literature in architecture is abundant, but it is neither widely believed nor followed. Hence, we have tried to give references that will both illuminate points and guide the interested reader to other useful works.

Because this is a book of essays and not a text, all the references and notes have been referenced in line with the text, and the reader is urged to follow the links for detailed study on the topic.

Chapter 1: Ambivalence of Multi-Cloud – Deploying a multi-cloud platform in an enterprise can have both advantages and challenges, and it ultimately depends on the specific needs and goals of the organization. In the chapter we will try to lay out details how deploying a multi-cloud platform in an enterprise can offer many benefits, but it also poses some challenges. Organizations should carefully evaluate their specific needs and goals before deciding whether to adopt a multi-cloud approach and should ensure that they have the necessary expertise and resources to manage multiple cloud environments effectively.

Chapter 2: Cloud Deployment Costs – The cost of cloud deployment can vary depending on several factors, such as the size of the infrastructure, the specific cloud service provider, the duration of usage, and the type of workload being deployed. The chapter analyzes are some of the main cost components that organizations should consider when deploying in the cloud. It is important to note that cloud deployment costs can be highly variable and are heavily dependent on the specific use case and requirements of each organization. To optimize cloud deployment

costs, organizations should consider carefully evaluating their infrastructure needs, choosing the right cloud service provider, optimizing resource utilization, and implementing cost management strategies such as monitoring, automation, and right-sizing.

Chapter 3: Security Sense of Cloud – The security of cloud computing is a topic of great concern for many organizations, particularly as more critical workloads and sensitive data are being migrated to the cloud. The chapter provides architectural views on some of the main security considerations for cloud computing. Overall, cloud computing can provide a high level of security when properly configured and managed. Organizations should work closely with their cloud providers to ensure that their applications and data are properly secured and that all necessary security controls are in place. Additionally, organizations should regularly review their cloud security posture and adapt to any changes in their risk profile.

Chapter 4: Availability and Disaster Recovery – Availability and disaster recovery are two related but distinct concepts in the field of IT infrastructure and operations. In this chapter we try to uncover these concepts in a practical approach. Availability is focused on preventing and minimizing downtime and disruptions, while disaster recovery is focused on recovering from major disruptions and restoring services as quickly as possible. Both are important considerations for ensuring the continuity and reliability of IT services, and organizations should have strategies in place for both availability and disaster recovery to ensure that they can continue to operate in the face of unexpected events.

Chapter 5: Cloud, Agile and Software Development Life Cycle – Cloud computing can be a valuable tool for Agile and Software Development Life Cycle (SDLC) processes, as it offers a range of benefits and capabilities that can improve development efficiency and flexibility. In the chapter we take through how cloud computing can be used in Agile and SDLC processes to make them even more efficient. Cloud computing can be a powerful enabler for Agile and SDLC processes, providing the flexibility, scalability, and automation capabilities required to support rapid and iterative development cycles. Organizations that adopt cloud computing as part of their development processes can gain a competitive advantage by accelerating the delivery of new features and functionality to their customers.

Chapter 6: Retrofitting Cloud Service Accurately – Retrofitting cloud services accurately involves a number of considerations and steps to ensure a successful

migration of existing applications or infrastructure to a cloud environment. In the chapter we talk about are some key considerations when architecting cloud services. Retrofitting cloud services accurately requires careful planning, thorough assessment, and meticulous attention to detail. By following proper steps, organizations can ensure a successful migration to the cloud, while minimizing disruption and maximizing the benefits of cloud computing.

Chapter 7: Design First then Code – Design first then code, is a software development approach that emphasizes the importance of planning and designing a project before starting the coding phase. It involves creating a detailed plan or blueprint of the project's architecture, functionality, and user interface, which can help to ensure that the final product is well-organized, efficient, and user-friendly. Once the design phase is complete, the coding phase can begin. By taking the time to plan and design the project before starting to code, developers can avoid common pitfalls such as scope creep, inefficient code, and user interface issues. This approach can also help to save time and resources in the long run, as it can help to identify potential problems early in the development process.

Chapter 8: Infra Team and App Team Becomes DevOps Team–The merging of an infrastructure team and an application team into a DevOps team can bring numerous benefits to an organization's software development process. We have taken and experienced this journey and through the chapter share the key advantages of creating a DevOps team. Creating a DevOps team by merging infrastructure and application teams can help to streamline the software development process, improve collaboration, and increase efficiency and flexibility, ultimately leading to better software products and a more competitive business

Chapter 9: Traits of Being a Good Cloud Solution Architect – A cloud solution architect is responsible for designing and implementing cloud-based solutions for businesses and organizations. Based on our experience with the best in the industry, we outline the traits that are important for a cloud solution architect to possess. A cloud solution architect can help organizations to achieve their goals by designing and implementing effective cloud-based solutions and work as technology leader for the cloud journey.

Chapter 10: Treat Data and Database Separately – It is important to treat data and database separately in software development. While the terms are often used interchangeably, they represent distinct concepts that require different approaches and considerations. We discuss the details along with advantages of this approach

in this chapter. Treating data and database separately can help to improve the quality and performance of software applications, while also making it easier to manage and maintain the codebase.

Chapter 11: Frozen Architecture is Obsolete Architecture – The concept of frozen architecture refers to an approach in software development where the architecture is designed to be static and unchanging over time. This approach is often contrasted with agile development methodologies, which prioritize flexibility and adaptability. While frozen architecture may have been a viable approach in the past, it is becoming increasingly obsolete in today's rapidly changing technology landscape. We detail out some reasons in the chapter. The concept of frozen architecture is becoming increasingly obsolete in today's fast-paced technology landscape. Instead, software development teams should focus on agile development methodologies that prioritize flexibility, adaptability, and responsiveness to changing business needs and emerging technologies.

Chapter 12: What Exactly is Software Architecture? – Software architecture refers to the high-level design and organization of software systems, which includes the structures of software components, their relationships, and the principles and guidelines governing their design and evolution over time. It is concerned with making strategic decisions about software systems that enable them to meet the functional and non-functional requirements of the system, such as scalability, maintainability, reliability, and security, among others. In this chapter we take you through the forgotten concept and emphasize why it is still very important to have Software architecture in today's agile world.

Coloured Images

Please follow the link to download the
Coloured Images of the book:

https://rebrand.ly/i9z0uhx

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

1. Ambivalence of Multi-Cloud 1
Introduction 1
Structure 1
Objectives 2
Multi-cloud computing 2
The Ambivalence 3
Multi-Cloud computing reference architecture 3
Simplifying Multi-Cloud computing architecture 5
Multi-Cloud governance and management 6
Cost management 7
Deployment 7
Migration 8
Disaster Recovery Management and Business Continuity Planning 8
ISO 22301 9
Security and compliance 10
Continuous auditing and compliance 10
Service management 11
Asset management 11
Monitoring and logging 11
Summarizing Multi-Cloud governance and management 12
Cloud usability and Multi-Cloud computing 13
Capable 15
Personal 15
Reliable 15
Secure 15
Valuable 16
Brain supplements 16
Domain-driven design and Multi-Cloud computing 16

Hub and spoke architecture – multi-Cloud computing 17
Conclusion 19

2. Cloud Deployment Costs 21
Introduction 21
Structure 21
Objectives 22
Modeling Total Cost of Ownership (TCO) for Cloud 25
Cloud TCO 26
Capital expenditure 27
Operational expenditure 28
Qualitative costs 29
Outage analysis and cost of outage 30
Cloud cost estimation and management 31
Using predictive analytics for Cloud cost estimation 31
Managing and monitoring Cloud deployment cost for cloud 32
Reference for Data Center TCO 32
Conclusion 33

3. Security Sense of Cloud 35
Introduction 35
Structure 35
Objectives 36
Data center security 36
Cloud security 37
Asset management 37
Access control 38
Cryptography 38
Physical and environmental security 38
Operations security 38
Communications security 38
System acquisition, development, and maintenance 38

Supplier relationships *39*
Cloud native tools versus specialized tools 39
Security reference architecture 40
Security Operations Center *40*
Network *41*
Servers *41*
Threat detection and protection *41*
Policy management *42*
Information protection *42*
Confidential computing *42*
Identity and access management *43*
Zero Trust Security (ZTS) 43
Zero Trust Network Access (ZTNA) *46*
Conclusion 46
Brain supplements 47

4. Availability and Disaster Recovery 49
Introduction 49
Structure 50
Objectives 50
High Availability (HA) vs Disaster Recovery (DR) 50
High Availability *51*
Disaster recovery *52*
Benefits of Business Continuity and Disaster Recovery 53
Principles of Business Continuity (BC) and Disaster Recovery (DR) 54
Securing regulatory body and organization leadership support *54*
Risk assessment *55*
Business impact analysis *55*
Business continuity plans *56*
ISO 22301 56
Making ISO 22301 effective *57*

Guidelines for Disaster Recovery and Business Continuity 57
Testing of BCP and DR drills..... 60
Review and improve your business continuity plan..... 61
Conclusion..... 61
References..... 61
IT Standards..... 61
ISO Standards and Business Continuity..... 62
USA BCP Standards for Financial Institutions..... 62
USA BCP Standards for Non-Financial institutions..... 62
USA Cross-Industry BCP Standards..... 63
USA BCP Standards for the Healthcare/Life Science Industries..... 63
Other National Standards..... 63
India..... 64

5. Cloud, Agile and Software Development Life Cycle..... 65
Introduction..... 65
Structure..... 65
Objectives..... 66
Introducing Cloud computing role in agile development..... 66
Development..... 67
Development environment..... 67
Parallel development and testing..... 68
Proof of Concept (PoC) and Research and Development (R&D) driven development..... 69
External system and platform availability..... 71
Testing..... 72
Test Driven Development..... 72
Test environment..... 73
Configuration management..... 74
Code repository branching..... 74
Continuous Integration and Continuous Deployment..... 75

Conclusion 75

6. Retrofitting Cloud Services Accurately 77

Introduction 77

Structure 77

Objectives 78

Decision to move to cloud 78

Business goals and objectives 79

Security and compliance 79

Cloud service provider 79

Migration strategy 79

Cost 79

Integration 80

Performance and scalability 80

Requirements and constraints for cloud migration 80

Principles for cloud migration 81

Approaches to cloud migration 83

Tools to use during cloud migration 84

Workload sensitivity 87

Non-functional aspects 87

Technical workload type 88

Technical capability requirement of workload 89

Integration 92

Operations and management 94

Cloud providers 98

Learnings 102

Conclusion 103

7. Design First then Code 105

Introduction 105

Structure 105

Objectives 106

Coding history 106
Design thinking 107
Approach to a practical problem 111
Design approach 112
Conclusion 119

8. Infra Team and Apps Team Becomes DevOps Team 121
Introduction 121
Structure 121
Objectives 122
DevOps 122
IT Team – A conventional look 122
DevOps – How to reform? 124
Cloud Operations Team – New Roles and Responsibilities 128
Program management - Updated responsibility 130
Conclusion 131

9. Traits of Being a Good Cloud Solution Architect 133
Introduction 133
Structure 133
Objectives 134
Role definition of Cloud Solution Architect (CSA) 134
Understanding business problems 135
Solution designing 137
Improve design with latest services 139
Define process and facilitate adoption 141
Conclusion 143

10. Treat Data and Database Separately 145
Introduction 145
Structure 145
Type of data 145

Data storage options....148
Processing of data....150
Conclusion....152

11. Frozen Architecture is Obsolete Architecture....153
Introduction....153
Structure....153
Objectives....153
Architecture versus design....154
Freezing an architecture....155
Business architecture....156
Data architecture....163
Application architecture....166
Technology architecture....170
Conclusion....173

12. What Exactly is Software Architecture?....175
Introduction....175
Structure....175
Objectives....175
Software architecture....176
Technology vision versus software architecture....176
Are enterprise architecture and software architecture the same?....177
Types of architecture....178
Monolithic architecture....178
User interface layer....178
Business logic layer....178
Data access layer....178
Application infrastructure....179
Third-party libraries....179
Deployment and release processes....179
Service-oriented architecture....179

Service 179
Service interface 180
Service registry 180
Service broker 180
Service bus 180
Service orchestration 180
Service composition 180
Service monitoring 180
Microservices architecture 181
Service 181
API gateway 181
Service registry and discovery 181
Load balancer 181
Containerization 182
Configuration management 182
Monitoring and logging 182
Database management 182
Event-driven architecture 182
Event producers 182
Event consumers 183
Event bus 183
Event store 183
Event processing 183
Event sourcing 183
Command Query Responsibility Segregation 184
Layered architecture 184
Presentation layer 184
Application layer 184
Domain layer 185
Data access layer 185
Infrastructure layer 185

Cross-cutting concerns *185*
Client-server architecture 186
Client *186*
Server *186*
Network *186*
Protocol *186*
Data storage *187*
Security *187*
Peer-to-peer architecture 187
Nodes *187*
Connections *187*
Routing *188*
Indexing *188*
Security *188*
Applications *188*
Components of software architecture 189
Components *189*
Relationships *189*
Patterns *189*
Quality attributes *189*
Constraints *190*
Views *190*
Documentation *190*
Roles in software architectures 190
Software architect *190*
Technical lead *191*
Software developer *191*
Quality assurance engineer *191*
DevOps engineer *191*
Database administrator *191*
Technical writer *191*