



Salesforce Architect's Handbook

A Comprehensive End-to-End
Solutions Guide

—

Dipanker Jyoti
James A. Hutcherson

apress®

Salesforce Architect's Handbook

**A Comprehensive End-to-End
Solutions Guide**

**Dipanker Jyoti
James A. Hutcherson**

Apress®

Salesforce Architect's Handbook: A Comprehensive End-to-End Solutions Guide

Dipanker Jyoti
Rockville, MD, USA

James A. Hutcherson
Orlando, FL, USA

ISBN-13 (pbk): 978-1-4842-6630-4
<https://doi.org/10.1007/978-1-4842-6631-1>

ISBN-13 (electronic): 978-1-4842-6631-1

Copyright © 2021 by Dipanker Jyoti and James A. Hutcherson

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484266304. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This book is dedicated to the Ohana of the Salesforce ecosystem.
Each person who contributes to its Trust, Customer Success,
Innovation, and Equality makes it possible to extend the ecosystem
to amazing heights.*

Table of Contents

About the Authors..... xiii

About the Technical Reviewerxv

Acknowledgmentsxvii

Introductionxxi

Chapter 1: Salesforce Architecture..... 1

 Why Do Companies Choose Salesforce?..... 2

 Modern-Day Options to Build Technology Solutions 3

 On-Premise Solution..... 6

 Infrastructure as a Service (IaaS) 7

 Platform as a Service (PaaS) 9

 Software as a Service (SaaS) 10

 Salesforce as a PaaS- and SaaS-Based Cloud Service Provider (CSP) 13

 The Lightning Application Layer 17

 The Seven Architecture Domains to Consider When Architecting with Salesforce 24

 Application Design Architecture 25

 Data Architecture..... 26

 Security Architecture 26

 Salesforce Integration Domain 27

 Salesforce Identity and Access Management (IAM) Domain 27

 Salesforce Mobile Domain..... 28

 Development Lifecycle Management 28

 Key Considerations and Limitations When Architecting with Salesforce 29

 Summary..... 29

Chapter 2: The Art of Artifacts..... 31

Art of Artifacts 32

FUSIAOLA Analysis 32

 F: Features..... 33

 U: User 38

 S: Systems..... 47

 I: Integrations..... 48

 A: Authentication 49

 O: Objects 50

 L: Licenses..... 51

 A: Assumptions 51

 Scope and Nature of Integrations..... 52

 Identity Provider–Initiated vs. Service Provider–Initiated Authentication 53

Format of the FUSIAOLA Analysis..... 54

System Landscape Artifact 54

 What Is the Salesforce Organization Strategy? 55

 What Are the Specialty Salesforce Licenses That You Will Be Using Beyond
 the Core Sales, Service, or Platform Licenses? 56

 What Are the AppExchange Products That You Will Leverage to Avoid
 Custom Development? 57

 How Many Different Types of Salesforce Community Licenses Are Needed for
 External (or Any Internal) Users? 58

 What Are the User Authentication/Authorization Tools for Both Internal and
 External Users to Authenticate into Salesforce? 58

 What Is the Integration Strategy to Integrate Salesforce with Existing or
 New Systems?..... 59

Data Model Artifact 61

 Objects Needed to Store and Manage Data in Salesforce..... 63

 Object Types 65

 Record Types Within Each Object 67

 Object Relationship to Other Objects..... 69

 The Organization-Wide Default Settings for Object Visibility..... 71

Owners of the Records Within Each Object	72
An Estimated Number of Records Per Object That Can Exist at Any Given Period	73
Summary.....	74
Chapter 3: Salesforce Application Architecture	77
Salesforce Licenses	77
Org-Level Licenses.....	78
User Licenses	80
Permission Set Licenses	85
Feature Licenses	86
Platform vs. CRM Licenses.....	88
Salesforce Application Design Choices.....	90
Common Declarative Options	91
Common Programmatic Options.....	96
Order of Execution Within Salesforce	101
Salesforce AppExchange.....	104
Considerations for Using Off-Platform Systems Instead of Salesforce.....	106
Salesforce Org Strategy Considerations	107
Chapter Summary	110
Chapter 4: Salesforce Data Architecture	113
Why Is Data Architecture Important?	115
Data Modeling in Salesforce	117
Data Model Techniques and Considerations.....	118
Top-Down Consideration with Bottom-Up Requirements.....	119
Cardinality: Salesforce Options.....	123
Selecting the Best Type of Relationship	123
Revisiting Cardinality in Salesforce.....	124
Other Data Modeling Considerations.....	125
Large Data Volume Architecture.....	127
What Causes LDV Issues?	127
Addressing LDV Issues	132

TABLE OF CONTENTS

- Data Lifecycle Management in Salesforce..... 134
 - Data Migration..... 135
 - Data Backup 144
 - Data Archiving 145
- Chapter Summary 146
- Chapter 5: Salesforce Security Architecture 147**
 - Why Is Security Architecture Important? 148
 - Three Pillars of Salesforce Security Strategy 149
 - Platform-Level Security: Access and Visibility..... 151
 - Sharing..... 153
 - Access to Data Behind the Scenes 164
 - Access Grants and Record Access Calculations..... 165
 - The Impact of License on Security..... 167
 - Platform Security 167
 - Standard (Out-of-the-Box) Salesforce Audit and Event Monitoring Capabilities 169
 - Data Encryption and Data Security Options..... 170
 - Salesforce Shield..... 174
 - Security for External User Experiences and Salesforce Portals 175
 - Using Groups to Manage Access and Visibility..... 175
 - Implicit Sharing Can Impact Record Sharing 176
 - Community Sharing for External Users..... 177
 - External Organization-Wide Defaults..... 177
 - Sharing Sets 178
 - Share Groups (for Internal Users) 178
 - External Role Hierarchy 178
 - Making Optimal Decisions 179
 - Chapter Summary 184
- Chapter 6: Salesforce Integration Architecture 185**
 - Integration Strategies 186
 - Integration Architecture Use Case 188
 - Integration Levels 192

Integration Layer	193
API Protocol Types	194
Middleware.....	197
Integration Choices	200
Overview of Integration Patterns	202
Approach	202
Timing.....	203
Source, Target, and Direction.....	203
Calling Mechanism	203
Error Handling and Recovery.....	205
Idempotent Design Considerations.....	205
Security Consideration	205
State Management	206
Integration Patterns.....	206
Integration Patterns, Uses Cases, and Justification.....	207
Selecting the Appropriate Integration Strategy and Integration Pattern.....	207
Request and Reply.....	208
Fire and Forget	210
Batch Data Sync	212
Remote Call-In.....	214
UI Update	216
Data Virtualization	217
Other Integration External Service Options	219
Chapter Summary	220
Chapter 7: Salesforce Identity and Access Management Architecture	223
Stages of an Identity and Access Management Lifecycle.....	224
Architecting Identity and Access Management in Salesforce.....	231
Provisioning.....	232
Access	233
Authentication	234
Authorization	238

TABLE OF CONTENTS

Session Management	244
Single Sign-On.....	248
Logouts and Redirects.....	251
User Account Management	253
Deprovisioning.....	253
Chapter Summary	254
Chapter 8: Salesforce Mobile Architecture.....	257
Why Mobile Architecture Is Important.....	257
It's a Lifecycle.....	258
Leverage Devices	260
Connects the World.....	261
Important Mobile Concepts.....	261
Modern Mobile Applications	261
User Experience.....	262
Access to Device Features	263
Speed to Market	263
Development Costs.....	264
Application Performance	265
Security (Device and API) Considerations.....	268
Development Skills.....	269
Deployment	269
Pulling It All Together	270
Using and Accessing Mobile Features	271
Instant Access to Applications and Data.....	272
Application Access (Authentication and Authorization)	272
Application Use.....	273
Mobile Security.....	273
Managing an Enterprise Mobile Environment.....	274
Enterprise Tools	274
Salesforce Mobile App Features	276
Application Design Considerations	277

Salesforce Mobile Development	279
New Salesforce App	280
Development Environment Differences	284
Salesforce Mobile SDK Development	288
Salesforce Supports IoT, Wearables, and Other Connected Devices	289
Understand IoT Architecture Patterns.....	290
Connecting Salesforce with IoT Devices	290
Implementing IoT on Salesforce	291
Chapter Summary	292
Chapter 9: Salesforce Development and Deployment Lifecycle.....	293
DevOps.....	294
Traditional Software Development Approach	295
Delivery Methodology	298
Waterfall Methodology.....	300
Agile Methodology.....	300
Hybrid Methodology.....	301
Salesforce Delivery Team.....	303
Environment and Release Management Strategy.....	308
Org-Based Development Approach	313
Source-Driven Development (SDD) Approach.....	314
Continuous Integration and Continuous Deployment (CI/CD) Strategy.....	318
Salesforce DX	318
Salesforce CLI.....	319
Scratch Orgs.....	319
Salesforce Package(s).....	320
Environment Hub	321
Testing Strategy	322
Governance and Monitoring.....	326
Steering Committee.....	327

TABLE OF CONTENTS

Center of Excellence (COE) 327

Data Governance Committee 328

Chapter Summary 329

Appendix A: Salesforce Authorization Flows..... 331

Web Server Flow 332

User-Agent Flow..... 333

JWT Bearer Flow 335

Device Flow..... 336

Asset Token Flow 338

Username and Password Flow..... 340

Refresh Token Flow 341

SAML Assertion Flow 342

SAML Bearer Assertion Flow..... 344

Appendix B: Salesforce Integration Patterns..... 347

Request and Reply 347

Fire and Forget..... 348

Batch Data Sync..... 349

Remote Call-In 350

UI Update..... 352

Data Virtualization 353

Appendix C: Salesforce Sample Artifacts 355

System Landscapes 355

Data Models 360

FUSIAOLA Options 362

Lifecycle Management Artifacts..... 362

Artifacts Used During a Mock CTA Review 364

Index..... 373

About the Authors



Dipanker “DJ” Jyoti is an industry veteran with over two decades of experience in managing digital transformation engagements with Fortune 500 companies. Dipanker holds a total of 23 industry certifications, among which 13 certifications are in Salesforce, including the Salesforce Certified System Architect and Salesforce Certified Architect certifications. Some of his other architect-level certifications include AWS Certified Solutions Architect - Professional, Google Cloud Certified Professional Architect, Microsoft Certified Azure Solutions Architect, and Certified Blockchain Expert.

Dipanker currently works for IBM as an associate partner, managing IBM’s Global Salesforce Assets. Previous to IBM, he held executive positions at Capgemini, Accenture, and Citibank. Dipanker’s involvement with Salesforce began a decade ago, as part of a strategic business transformation engagement, where he got intimately involved in building a Salesforce proof of concept to demonstrate the capabilities of using a low code/no code CRM solution. He instantly fell in love with Salesforce and has since continued his journey to grow and share his knowledge of Salesforce among his peers and the cloud community. He currently manages an online blog on cloud architecture concepts, accessible at www.cloudmixdj.com. Dipanker lives with his wife, Junko, and two sons, Kazuya and Ouji, in Rockville, MD. In his free time, he enjoys cooking for his family and watching premier league soccer. For more details about Dipanker, you can follow him on LinkedIn at www.linkedin.com/in/dipanker-dj-jyoti-3104974.

ABOUT THE AUTHORS



James A. Hutcherson is a seasoned business leader with more than 40 years of technology experience. His first Salesforce project was implemented in 2004 where he started his Salesforce journey. During this time, James had taken every opportunity to grow his Salesforce knowledge. He holds 17 Salesforce certifications including Certified Salesforce Application Architect and Certified Salesforce System Architect. James is an avid educator and has hosted several *no-cost* Salesforce training sessions over the last few years to give back and grow the Salesforce ecosystem.

James is currently a principal and the chief technical architect at Capgemini Government Solutions where he leads the Salesforce Practice delivering enterprise-class solutions to US federal government customers. He earned an MBA from Baldwin Wallace College, an MSCIS from the University of Phoenix, and a BA from Southern Illinois University - Carbondale. James and his wife, Sherry, live in Orlando, FL, where they enjoy spending time with their family and four wonderful grandchildren. Follow him on LinkedIn at www.linkedin.com/in/jim-hutcherson-2018/ and Twitter at <https://twitter.com/hutchersonj>.

About the Technical Reviewer



Kal Chintala is one of about 300 Salesforce Certified Technical Architects (CTAs) worldwide with over 15 years of experience in delivering Salesforce and other CRM systems architectures, software design, and data modeling. Most of these projects are executed in Agile fashion while managing technical risk and change and software quality assurance for enterprise-wide transformation of many federal and Fortune 500 companies. Kal has managed Salesforce architectures for up to 65,000 users globally, including achieving more than \$1.2 million in operational efficiencies by consolidating services onto Salesforce. His demonstrated industry experience achieves operational efficiencies through innovative applications of technology and business process enhancements. Kal currently serves as a Salesforce Practice strategy advisor, a technology lead, and an SME on road map and delivery of complex enterprise-wide Salesforce implementations across federal civilian and defense clients. Kal earned his bachelor's from Madras University, India, and has a master's from La Trobe University, Australia. He has earned more than 25 certifications including multiple certifications in Salesforce. Kal lives with his wife and two beautiful kids in Ashburn, Virginia. You can follow him on LinkedIn: www.linkedin.com/in/kalch/.

Acknowledgments

A book like this cannot be accomplished without the support and compromise made by my family. My beautiful and amazing wife, **Junko Maki**, who has spent the last 9 years of her life unconditionally loving me and selflessly supporting my ambitions. She is my best friend, philosopher, coach, and therapist. The last part is probably something I need a lot of. Her understanding and compromise to allow me to spend several nights and weekends in my office room by myself is an equal contribution made by her in completing this book, as it has been for me. Thank you for all you do for me and the kids every single day, in addition to your real estate career. I could not have been where I am without you in my life.

I am also the proud father of two teenage sons, **Kazuya Jyoti** and **Ouji Jyoti**, who were the first reviewers of the initial chapters of this book. My efforts were already paid in full the day they both read the first chapter of this book and said they really want to become cloud architects. Their persistent request to write more chapters so they could learn more about Salesforce kept me going with a relentless motivation to research and write each chapter with love and dedication.

Giving back to the community is something I learned at a young age by simply observing my mother, **Arati Jyoti**, who is a Reiki Grandmaster and a volunteer at Mother Teresa's order of missionaries and St. Jude Children's Hospital in Mumbai, India. She dedicated the last 30 years of her life to offering Reiki treatments to people with various physical, emotional, and mental illnesses. In addition to this, she volunteers her remaining time to St. Jude Children's Hospital. My father, **Haran Jyoti**, who has selflessly dedicated his life to meeting his family's every need and raising his only son with the belief that everything is achievable as long as you don't give up. He was the person to tell me "you can do it" when I said I wanted to write a book on Salesforce architecture.

I am truly blessed to have my friend and co-author, **James Hutcherson**, partnering with me on this book, without whom this book could not have been completed. Thanks for trusting me and diving in with all your dedication. Your collaboration and contribution has been truly invaluable.

ACKNOWLEDGMENTS

I want to thank **Matt Francis** for his constant support and feedback on this book. Matt is a Salesforce veteran and a Salesforce Certified Technical Architect (CTA), who has not only been a Salesforce mentor to me but has been an unstoppable force in my success with my Salesforce career. His nonstop encouragement and endless motivation each day at work has led my Salesforce journey to be where it is today.

A book is only as good as its reviewer, for which I want to thank **Kal Chintala**. His invaluable knowledge in Salesforce in general and as a Salesforce Certified Technical Architect (CTA) has helped us tremendously in correcting many parts of this book and getting new perspectives on our content.

Similarly, a great contribution was made by the Salesforce Generations CTA group that reviewed our book and provided their unbiased review for our material. The Salesforce Generations CTA group is a weekly CTA study group that includes **Sudhir Durvasula, Waruna Buwaneka, Chetan Devraj, Brock Elgart, Jim Hutcherson, and Julia Kantarovsky**, in addition to James and me. We dedicated several of our study group sessions to reviewing each chapter line by line and getting collective feedback from the group.

Last but not the least, I would like to thank **Rita Fernando** and **Susan McDermott** from Apress for believing in me and giving me an opportunity to write this book for Apress. I still remember the day I contacted both of you, and never in a thousand years could I have imagined what this journey was going to be. Thanks for your patience and continuous support throughout the entire process.

—**Dipanker “DJ” Jyoti**

I would like to thank my adorable wife, **Sherry Hutcherson**, for her loving support along this journey. She is my best friend and has been my champion for more than 40 years. She allowed me to accomplish more than I ever could have done without her giving me advice and inspiration. Her encouragement made my contribution to this book a reality. Thank you, Sherry, for everything you do and for allowing me to have and obtain the “big dreams.”

I am honored that **Dipanker “DJ” Jyoti** invited me to join in this endeavor. Without all the hard work you invested from start to finish, it would still just be a thought in my head. Your hard work getting this book approved and supported by **Apress** made it a reality. Thank you for letting me join your effort to write this book.

I would like to thank the members of the Salesforce Generations weekly CTA study group including **DJ, Sudhir Durvasula, Waruna Buwaneka, Chetan Devraj, Brock Elgart, Jim Hutcherson** (my son), and **Julia Kantarovsky**. They have provided thought leadership and assistance for both studying for the CTA exam and allowing DJ and me to ask hard questions and test topics with seasoned professionals in the field.

I am grateful to my family for the words of encouragement and the support over the years. My son, **Jim Hutcherson**, has shown me that even an *old dog* like me can still learn new tricks. You are an amazing Salesforce architect. But more importantly, you are the best son I could have ever asked for. You are an exemplary father, husband, and role model for everyone you meet. **Maureen Hutcherson**, thank you for the love, support, and fun you bring to the family. **Dave and Vickie Magill, Kathy Bevec, and Patty and Ronnie Eamich**, thank you for accepting me even when I'm "preoccupied" and still providing assistance and encouragement at just the right times. Thank you to **Renee and Richard Abell** and **Chis and Jim Sabo** for cheering me on over the years. You helped more than you will ever know.

I would not have co-written this book without the opportunities, challenges, and successes I gained from the amazing organizations I have worked with over the years, including **Capgemini, Adnubis, Apollo Education, Net Direct, Advizex, Agilysys/ Pioneer-Standard, NCR, Anderson Jacobson**, and the **US Navy**. My architectural knowledge evolved and matured from each of you allowing me to try, succeed, and sometimes fail. I would like to make a shout-out to **Capgemini Government Solutions** for investing actively in my career and the larger Salesforce ecosystem. Thank you.

I would like to offer a special thank you to **Kal Chintala** for his tireless support and guidance as the technical reviewer for this book. Your comments and ideas truly helped make it more valuable to our readers. Thank you for the effort and your friendship. It is greatly appreciated.

Finally, I would like to offer my sincere gratitude to **Rita Fernando, Susan McDermott**, and the staff at **Apress** who saw the value of this material and partnered with **DJ** and me in its publication. You have consistently provided the highest level of support, directions, and patience throughout the entire writing process, and for that you have my greatest appreciation.

—James A. Hutcherson

Introduction

This book is intended for several different audiences including (1) Salesforce implementation partners who are building their practice and want to define the overall strategy and architecture approach, (2) Salesforce industry professionals who need a one-stop educational resource to Salesforce architecture and for the Salesforce architecture domain certification exams, and (3) enterprise cloud champions who are involved in implementing, optimizing, and architecting Salesforce-based solutions within their organization.

What This Book Is About

This book takes an in-depth look into the seven key architecture domains that form the pillars of any Salesforce-based solution.

Platform as a Service Architecture

- Provides an overview of the Force.com platform as a service (PaaS) architecture and the key components that empower the Salesforce SaaS model.
- Understand the microservices-based architecture of Salesforce Lightning consisting of highly scalable and reusable Salesforce Lightning components, Salesforce Lightning pages, and Salesforce Lightning apps.
- Understand “When to Use What” standard features of Salesforce and the considerations, limitations, and trade-offs of declarative vs. programmatic configurations.
- Understand the AppExchange marketplace and the significance of considering third-party AppExchange microservices components and AppExchange apps.
- Understand the considerations, limitations, and trade-offs between declarative and programmatic design on the Salesforce platform.

Salesforce Data Architecture

- Provides an overview of the Salesforce platform architecture considerations when working with large data volumes (LDVs).
- Understand the object-oriented data modeling concepts within Salesforce and key considerations in designing the right Salesforce object model.
- Understand the data migration strategy, considerations, and appropriate tools to use.

Salesforce Security Model Architecture

- Understand the “Who Sees What and How” within Salesforce.
- Architect a solution that utilizes the appropriate platform security mechanisms.
- Design a secure Salesforce community portal architecture including access by both internal and external users.
- Identify declarative platform security features that can be used to meet record-level security requirements.
- Identify the programmatic platform security features that can be used to meet security requirements.
- Describe how to incorporate the platform security features into a solution.

Salesforce Integration Architecture

- Understand and evaluate the integration strategy for cloud-to-cloud, cloud-to-on-premise, and multi-Salesforce org integration scenarios.
- Understand when to use Canvas apps vs. integrating with Heroku apps vs. integrating using Salesforce Connect vs. integrating using a middleware such as MuleSoft.
- Understand the various integration patterns and justify their use as part of the overall integration architecture.

- Recommend and justify the appropriate integration strategy and integration patterns.
- Recommend the appropriate platform-specific integration technology.

Salesforce Identity and Access Management Architecture

- Understand the concepts of identity provider (IDP)-initiated and service provider (SP)-initiated protocols.
- Understand the out-of-the-box declarative capabilities to configure single sign-on (SSO).
- Understand the capabilities of integrated apps and ODATA integrations.
- Understand the benefits, considerations, and trade-offs of incorporating external applications.
- Understand the considerations, trade-offs, and benefits of using middleware solutions vs. Salesforce Identity.

Mobile Architecture

- Understand the concepts of Mobile Device Management (MDM) and Mobile Device Federation (MDF).
- Understand the considerations for a mobile platform strategy.
- Understand the Salesforce mobile architecture options, design trade-offs, and benefits.
- Understand the design approaches of architecting a Salesforce mobile solution using Mobile SDK or Salesforce1 Mobile App or the Field Service Lightning app.
- Understand the capabilities of the Salesforce Mobile SDK.
- Understand the strategies and use cases for wearables and connected devices and the appropriate architecture patterns.

Development and Deployment Lifecycle

In addition to the six architecture concepts outlined previously, this book also offers the industry best practices and the recommended framework for approaching, managing, delivering, and continuously improving a Salesforce solution using the Salesforce Development and Deployment Lifecycle:

- Understand the Salesforce Development Lifecycle Methodology (SDLM).
- Learn to evaluate and mitigate Salesforce implementation risks, dependencies, and constraints.
- Understand the key stakeholders needed for any Salesforce projects and understand their roles and responsibilities.
- Understand the best practices for setting up a Center of Excellence and a Salesforce Governance Model.
- Identify the appropriate test strategy and user acceptance strategy for Salesforce implementations.
- Learn to evaluate and use the appropriate tools for Salesforce project management, business requirements management, testing, and release management.
- Understand benefits and best practices for using source control and continuous integration for Salesforce release management.

What This Book Is *Not*

This book is not a screen-by-screen configuration guide for Salesforce, nor does it have any code snippets to learn programming. Although configuration, Apex, and Visualforce capabilities are discussed in this book, they are explained at the level that is necessary for an architect to architect an ideal Salesforce solution.

This book is not an encyclopedia covering every feature of Salesforce as we did not intend to replace the user manuals and Salesforce Trailhead¹ published by Salesforce. We have made the best attempt to add as many references as possible to Salesforce content, Salesforce trailheads, and other Salesforce contents throughout the book. Even if we could have covered every feature of Salesforce, it would be outdated by the time you read this book, given the pace at which Salesforce changes every day.

Instead, this book covers the consistent context needed by a Salesforce architect to group old and new features for their architectural use. It also connects the dots between any two features by categorizing their relevance within the seven architectural domains.

How Best to Read This Book

We have arranged each chapter in this book in the most logical order possible to match the sequence of architecting an ideal Salesforce solution. We have decoupled each chapter from other chapters such that they can be read independently on their own. For the readers who are just beginning their architect journey, we would definitely recommend completing Chapters 1 and 2 before jumping into any domain-specific chapters located within Chapters 3–9.

For some of our advanced architects, we would highly recommend starting with Chapter 2 before jumping into any domain-specific chapters located within Chapters 3–9.

This book is meant to be an architecture handbook that should be referenced and revisited by Salesforce architects throughout their architectural practice. For instance, if you are working on an assignment which requires heavy integrations, you can jump directly to Chapter 6 or Appendix B to get everything needed to design your integrations.

The content in this book is a culmination of our combined industry experiences and consultations with Multiple Salesforce Certified Technical Architects (CTAs) and aspiring CTAs in their approach to architecting a Salesforce solution. The content in this book comprises the learning materials we used to achieve our Salesforce Certified Application Architect and Salesforce Certified System Architect certifications. This book is also a reference guide to studying for the Salesforce CTA board exam.

¹https://help.salesforce.com/articleView?id=mth_what_is_trailhead.htm&type=5.

CHAPTER 1

Salesforce Architecture

You have been assigned to build something in Salesforce. You may have been an admin or developer or merely heard about Salesforce. Where do you begin?

You probably started by searching on YouTube, Google, or even Trailhead, Salesforce's free learning management tool, to learn everything about Salesforce, but all you found was piecemeal information about Salesforce and its architecture. There are several books on administration and development with Salesforce, but what about architecting with Salesforce? Here we will start with the fundamentals of Salesforce architecture and what makes it unique compared to other technologies.

This chapter covers some key points, including

- Why companies choose Salesforce over other cloud solution alternatives
- Distinctions between on-premise, infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) options to build solutions
- How Salesforce can be used as a PaaS as well as a SaaS solution
- The mechanics of a multi-tenant architecture and the metadata-driven framework that drives Salesforce
- The seven architectural domains of Salesforce that need to be considered when architecting with Salesforce

Why Do Companies Choose Salesforce?

Salesforce was started in 1999 by Marc Benioff, Parker Harris, Dave Moellenhoff, and Frank Dominguez.¹ In its 20 years of exponential growth, Salesforce has surpassed a market cap of more than \$180 billion² with a total annual revenue of \$13.3 billion in the year 2019 and employing over 36,000 global employees.³ The annual revenue for the year 2020 is estimated to surpass \$16.5 billion, which is over \$3 billion in revenue growth within a single year (see Figure 1-1).⁴

Salesforce’s Exponential Growth (as of 2020)

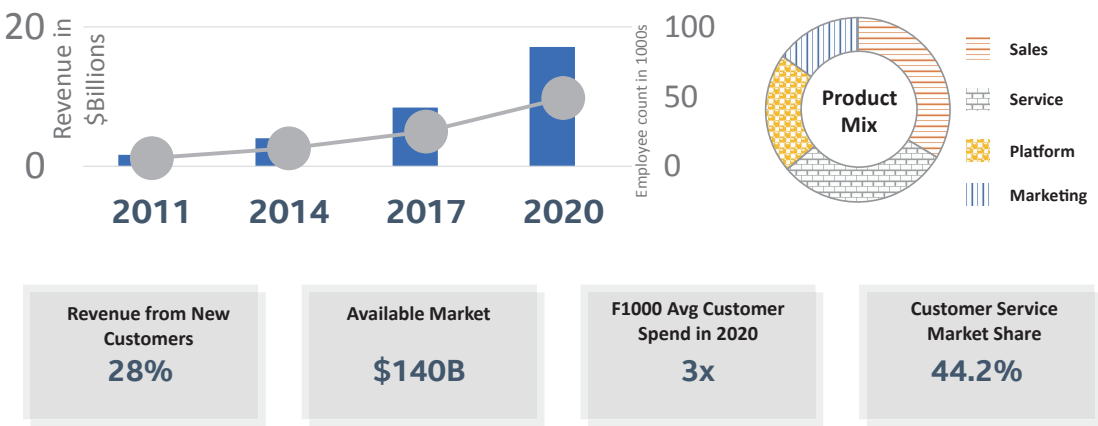


Figure 1-1. Salesforce Growth Over the Years

¹“The History of Salesforce.” Salesforce News, September 18, 2020, www.salesforce.com/news/stories/the-history-of-salesforce/.

²Jordannovet. “Marc Benioff’s Salesforce Has Eclipsed Larry Ellison’s Oracle in Market Cap.” CNBC, July 10, 2020, www.cnbc.com/2020/07/10/salesforce-eclipses-oracle-in-market-cap.html.

³“2019 Annual Report: Celebrating 20 Years of Salesforce.” www.annualreports.com/HostedData/AnnualReportArchive/s/NYSE_CRM_2019.pdf.

⁴“The History of Salesforce.” Salesforce News, September 18, 2020, www.salesforce.com/news/stories/the-history-of-salesforce/.

Salesforce provides several benefits over other cloud solutions, including

1. **No Code/Low Code Platform:** Most things in Salesforce can be configured by a business user with no programming skills. Almost everything in Salesforce is configured using and by just using a GUI interface with drag and drop capabilities that support reusable components.
2. **Time to Market:** Building production-ready business applications in Salesforce can be achieved within days, not months.
3. **No Additional Costs for All Upgrades Made to the Platform Each Year:** Salesforce constantly improves its SaaS and PaaS offerings by releasing new features three times a year that are automatically released to all Salesforce customers at no additional cost. The three major upgrades each year keep the platform in sync with the latest technology features in the industry, such as upgrades related to artificial intelligence, block chain, and machine learning.

Modern-Day Options to Build Technology Solutions

Before we begin architecting in Salesforce, it is important to understand all the options out there to build a solution and when building something in Salesforce makes sense.

Every solution requires a technology stack to operate in. A typical technology stack consists of eight tiers:

1. Application code
2. Runtime engine
3. Integration server
4. Operating system (OS)
5. Virtualization engine
6. Network device
7. Computation server
8. Data storage server

In this book, we will not get into the details of each tier, as each tier merits its own book. But for our context, it's important to realize that an organization needs to invest in all eight tiers to build even the simplest of solution. Whether they build one or multiple solutions, the eight tiers are the required building blocks. For any organization, investing in these eight tiers can range from a few thousand dollars to millions of dollars.

Given the investments needed, every organization has four options to build solutions. They are as follows:

1. Invest in an on-premise solution.
2. Leverage an infrastructure as a service (IaaS) cloud solution.
3. Leverage a platform as a service (PaaS) cloud solution.
4. Leverage a software as a service (SaaS) cloud solution.

The four approaches listed here differ from each other based on which tiers a company manages on their own vs. which tiers the company outsources to an external cloud service provider (CSP) such as Salesforce, ServiceNow, Appian, AWS, Microsoft Azure, Google, and so on.

In Figure 1-2, I have outlined the four options in a side-by-side comparison based on who manages which tier.

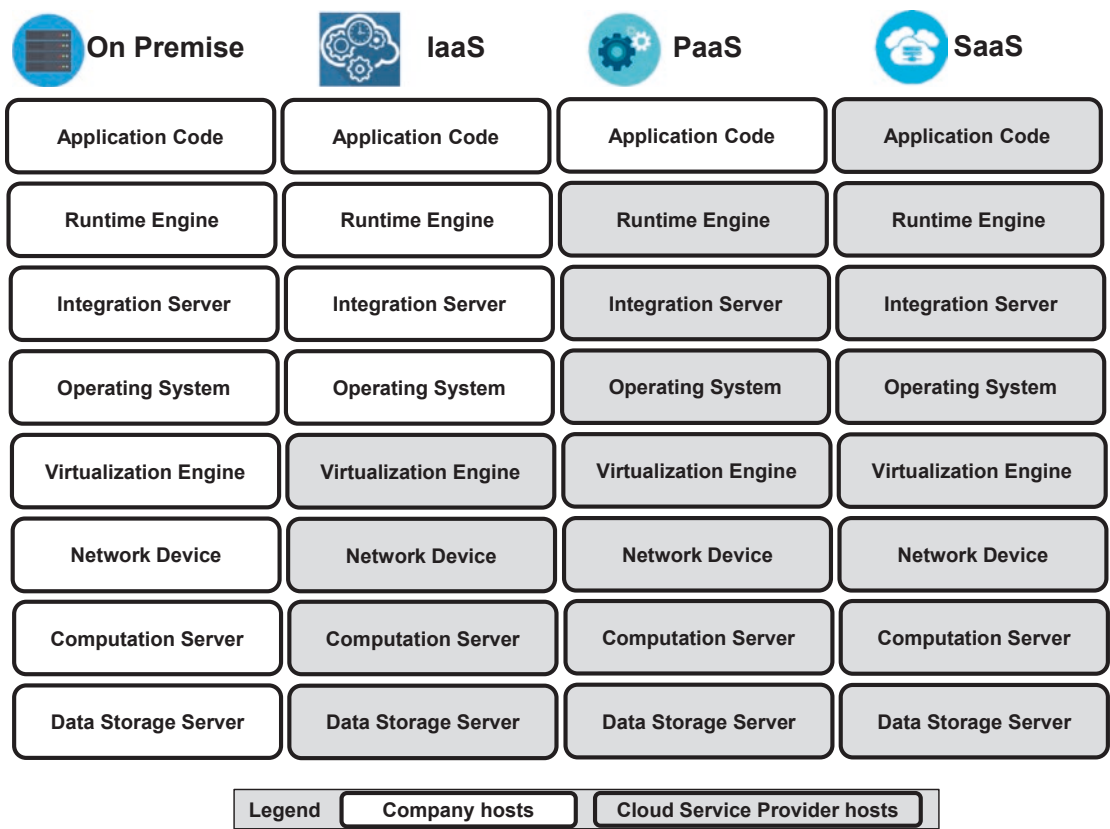


Figure 1-2. Comparison of On-Premise, IaaS, PaaS, and SaaS⁵

The various technology options can also be viewed in terms of a nested diagram with each option being an improvement over the other.

In Figure 1-3, I have illustrated such a nested diagram with a few examples of the popular CSPs that provide each type of service.

⁵<https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas>

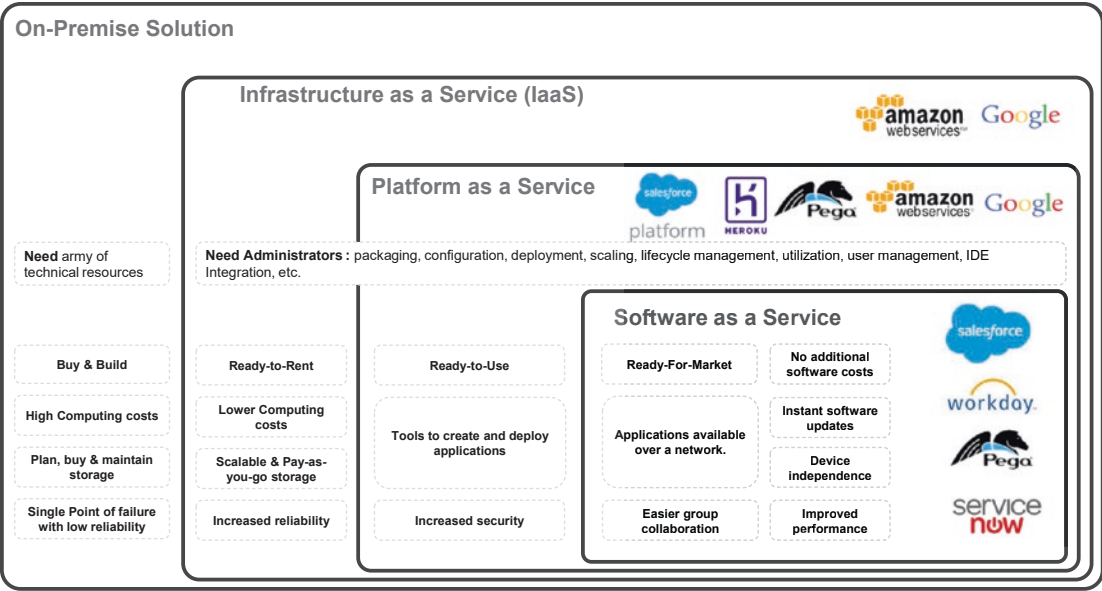


Figure 1-3. Nested Diagram of Solution Development Options

On-Premise Solution

Traditionally, any company interested in building a business application had to size up, buy, and set up all the tiers of the technology stack in-house from scratch. They would also need to manage and maintain each tier set up within the organization’s premises, *hence the industry term “on-premise.”* With this approach, the organization requires dedicated staff to manage the technology infrastructure, often at a stack-by-stack (*i.e., each infrastructure component*) level.

Some key advantages and disadvantages of the on-premise option are provided in Table 1-1.

Table 1-1. *Advantages and Disadvantages of On-Premise Solution Development*

Advantages	Disadvantages
Full ownership and control of the technology stack used to support the business operations. Every tier of the stack can be custom designed and managed to meet the unique needs of the organization.	Higher initial capital outlay and cost for specialty resources. On-premise solutions are expensive to build, and a single company cannot have economies of scale to support resources and component cost compared to other CSP solutions.
Internally administered security for the organization’s data, application code, and user base is proprietary to – and managed entirely within – the organization. Access to the entire technology stack resides within the organization’s premises. Each technology component can be custom tailored or entirely replaced to meet the organization’s needs for adherence to security standards.	Time intensive to build with slower time to market. It is time consuming to plan, size up, and procure the hardware and software needed. The initial planning also does not consider seasonal spikes and load demands. Additionally, resources at each tier need to coordinate and self-organize their divisions to support the solution, which could take months or even years to launch a single solution in production environment.
Controlled solution deployment can be entirely self-managed and timed as per the availability of in-house technical staff and without any reliance on third-party cloud providers because each component of the technology stack can be designed, tested, approved, and launched for live production use through internal coordination and readiness.	Difficult to keep up with industry standards in security and best practices. The technology industry is changing at a rapid pace with new standards emerging on a daily basis. Some standards may be nice-to-haves for a company, but some may be critical to compliance and market adherence. Failure to comply with these standards could even mean closing down business completely.

Infrastructure as a Service (IaaS)

In an infrastructure as a service (IaaS) model, cloud service providers such as Amazon Web Services, Microsoft Azure, or Google Cloud Platform provide outsourced raw computing infrastructure, data storage, and virtualization that are provisioned on demand within minutes and can be available instantly.

Some advantages and disadvantages of the IaaS approach are provided in [Table 1-2](#).