



# SYSTEM ARCHITECTURE



# Table of Contents

<b>Copyright .....</b>	<b>3</b>
<b>The Salesforce Architect Domain .....</b>	<b>4</b>
<i>Prerequisites and the Continued Journey.....</i>	<i>4</i>
<i>Isaac and Amaya.....</i>	<i>4</i>
<b>Universal Healthbit .....</b>	<b>5</b>
<b>Acme Insurance .....</b>	<b>6</b>
<b>BabyBox.....</b>	<b>7</b>



# Copyright

© Copyright 2000-2017 salesforce.com, inc. All rights reserved. Various trademarks held by their respective owners.

This document contains proprietary information of salesforce.com, inc., it is provided under a license agreement containing restrictions on use, duplication and disclosure and is also protected by copyright law. Permission is granted to customers of salesforce.com, inc. to use and modify this document for their internal business purposes only. Resale of this document or its contents is prohibited.

The information in this document is subject to change without notice. Should you find any problems or errors, please log a case from the Support link on the Salesforce home page. Salesforce.com, inc. does not warrant that this document is error-free.



# The Salesforce Architect Domain

## Prerequisites and the Continued Journey

The Salesforce System Architecture domain comprises the more programmatic designer domains:

- Integration Architecture
- Identity and Access Management
- Mobile Strategy Architecture
- Development Lifecycle and Deployment and
- Apex and Visualforce

At the designer level, an architect masters skill sets and concepts specific to each domain. However, at the domain level, the architect is expected to utilize these disparate skills and concepts almost like building blocks to design and articulate an optimal architectural solution for enterprise business customers.

## Isaac and Amaya

In this course we will follow Isaac, a certified Technical Architect, as he mentors Amaya, a program architect hoping to develop the expertise necessary to maneuver the complex landscape of System Architecture.

The pair will tackle two hypothetical scenarios that mirror real life architectural challenges. In the first scenario, Isaac outlines the specific steps an architect should follow in order create an optimal solution that addresses all of the scenario's system requirements.



## Universal Healthbit

Universal HealthBit (UHB) is a US company that provides monitoring devices and services for people with diabetes. Their skin-based device measures insulin levels in real time throughout the day without requiring a blood sample, saving their users more time and providing more convenience. Additionally, the device pairs with a smartphone via Bluetooth and saves real-time measurements to the UHB portal.

UHB's product has become wildly popular, and they have experienced tremendous growth in their 3 years of business, now serving 1.5 million customers. UHB anticipates serving 50% of the market (currently 30 million people with diabetes in the US) in the next 5 years. However, UHB's current systems and processes will not scale well, and they have identified processes they'd like to move to the Force.com platform.

Click here to begin the System Architecture digital training:

[Module 1: Hypothetical Scenario](#)

[Module 2: Gather and Define the System and Integration Requirements](#)

[Module 3: Recommend Off-Platform Components and Licenses](#)

[Module 4: Define the End-State System Landscape](#)

[Module 5: SSO and Social Sign-On](#)

[Module 6: Solution Design and Justification](#)

[Module 7: Project Management / Governance](#)

[Module 8: Implementation Strategy](#)



## Acme Insurance

In this second scenario, Amaya takes the reins in addressing the system requirements set forth in the use case. While Isaac still acts as her mentor, Amaya is much more versed in the steps it takes to diagram and present a viable solution to business stakeholders. Amaya also discovers alternative methods of fulfilling the business requirements—a testimony to her understanding of the nuances of programmatic architecture.

Click here to continue the System Architecture course:

[Module 1: Hypothetical Scenario](#)

[Module 2: Gather and Define the System and Integration Requirements](#)

[Module 3: Recommend Off-Platform Components and Licenses](#)

[Module 4: Define the End-State System Landscape](#)

[Module 5: SSO and Social Sign-On](#)

[Module 6: Solution Design and Justification](#)

[Module 7: Project Management / Governance](#)

[Module 8: Implementation Strategy](#)



## BabyBox

Having observed Isaac and Amaya present comprehensive architectural solutions to two rather complicated business use cases, it is now your turn to try your hand at deconstructing a hypothetical scenario and designing an optimal programmatic solution that includes a proposed system landscape diagram.

Click [here](#) to download the BabyBox hypothetical scenario (password oprlxmfg).

Click [here](#) for the System Architect Checklist that will aid you in your solution process (password TscCokUL).

Now that you have outlined a detailed solution to the BabyBox business requirements and created a diagram of your proposed end state system landscape that includes integration callouts and authentication flows, please click [here](#) for the Solution Key (password ctsIwzrJ) and [here](#) for the suggested System Landscape diagram (password IXDoubVS).

Remember that Amaya discovered that some system requirements had more than one optimal solution—that there was often two or three process that would fulfill them. When reviewing the Solution Key and comparing it to your proposed solution, please keep this in mind.

If you would like to discuss your proposed solution with your System Architect peers, please join the Salesforce Architect Success Group in the Salesforce Success Community. Here you will be able to bounce your ideas off of fellow architects who have also designed their own solutions to the BabyBox hypothetical scenario.