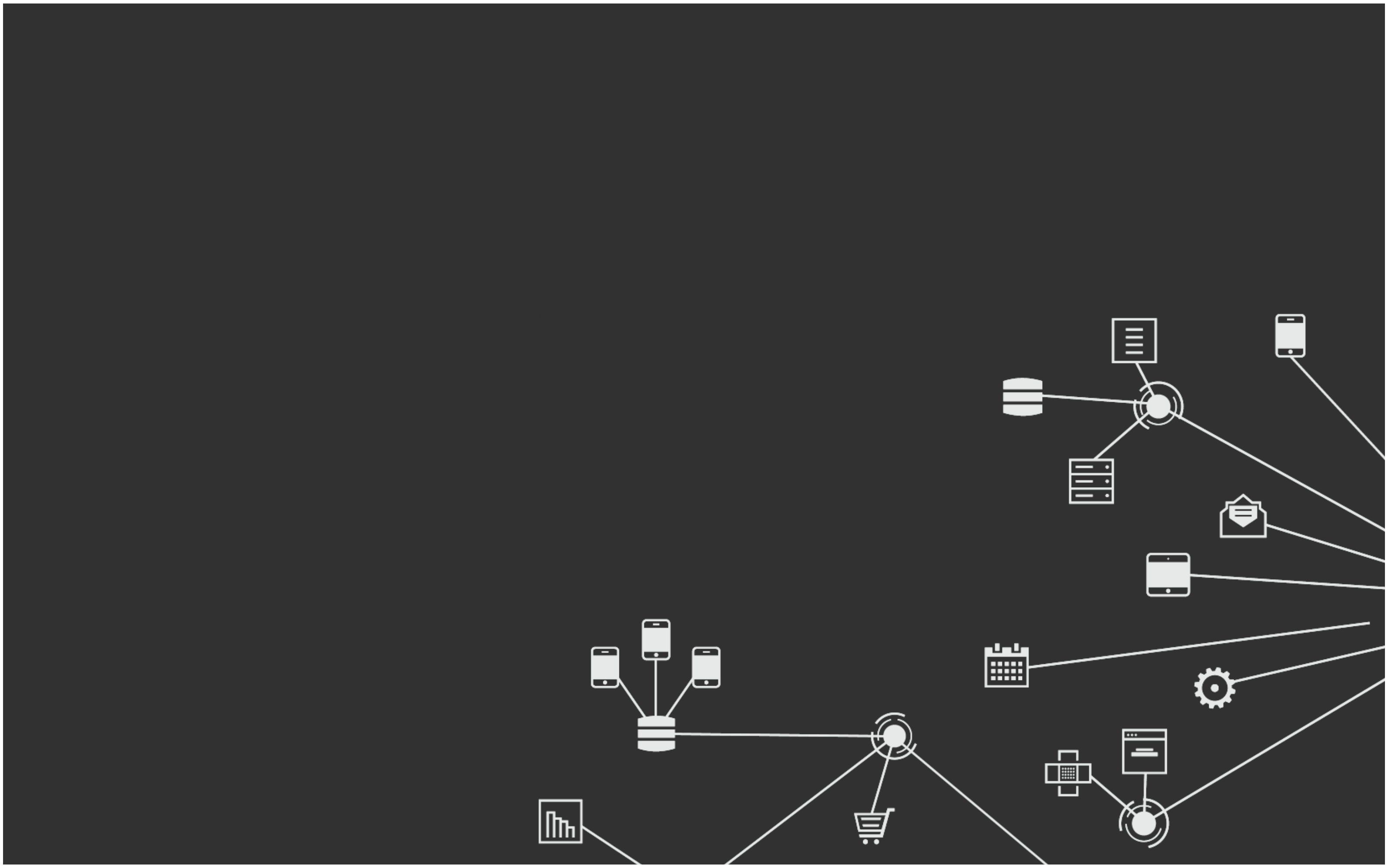
Anypoint Platform Architecture Solution Design

Companion Workbook



This course came to be through the creativity and hard work of many people:

Ryan Denney
Jeanette Stallons
Ryan Hoegg
Ryan Dodds
Roy Prins

Reflections

Each module includes a number of Reflections, which are questions meant to help you apply the ideas to your work and experience. These are not meant to be part of an assessment; instead, use them as the basis for conversations with colleagues, prompts for further study, or simply an aid to help you frame the concepts using your own experience.

Read the reflections as you start each module, and consider them as we work through it. Write your responses any time you feel inspired. At the end of the module, we'll set aside a few minutes to reflect individually and then discuss them with our groups.

Exercises

The context for the exercises is the digital transformation of a fictional bank, the Acme Bank. You will be defining the architecture of the new digital platform, an important initiative of the new CIO. The platform will enable the bank to migrate their legacy systems and expand their business by enabling developers to quickly build new components and integrations.

Prerequisites & requirements

For these exercises, you will be creating architectural blueprints and designs. These can be written directly in this workbook or you may use a word processor and/or diagramming tool that you are comfortable with. Since you will be working with a group, determine how you will share your work so that each member has access to it after the class.

There are freely available online drawing tools such as draw.io and lucidchart or commercial diagramming tools such as Visio or Omnigraffle which provide stencils or templates for diagramming.

Module 1

Enterprise Integration

Reflections

Are things in IT changing faster now than they used to? Is this accelerating? Why?

How does the increasing complexity of enterprise IT and enterprise integration affect my work?

How do big middleware and point-to-point integration impact project risk and delivery speed?

Are there obstacles to moving toward continuous delivery of integration software? What can you do about them?

Does the application network approach to integration make sense for your organization? Why or why not?

Has the organisational structure of the teams doing "integration" been changing in recent years? If so, how?

Scenario Background

Acme Bank's heterogeneous IT landscape consists of a variety of systems, applications and databases. The systems lack proper integration, though there is some point-to-point integration for specific purposes.

It recently became clear that the current landscape does not scale and does not provide enough flexibility and capacity to expand. It also becomes increasingly difficult to maintain some old legacy systems.

The newly appointed CTO's main responsibility is to lead Acme Bank into the digital era by providing new and innovative applications and functionalities. The focus will be on quickly enabling web based and mobile applications, which are yet to be developed. Also, in due time several legacy systems are to be replaced by custom Java applications. To that end, a new digital platform will be created.

The platform will allow development of new services and the integration of old and new resources (systems, applications, databases etc).

Exercise 1

Architectural Blueprint Rationale

During just one month on the job, the new CTO has worked with the CEO and executive team to understand the business plan, and each department's current projects, requests, concerns, and frustrations from IT. She also learned each part of IT's current workload, concerns, and plans.

She has now asked the architecture team (you) to present the rationale on which architecture decisions are based at Acme. It is no secret that she believes a good architecture practice is critical to the success of the digital transformation initiative, and good architectures emerge from consistency in the goals and constraints we use while creating them.

Determine the goals you believe should inform the architecture for Acme's digital transformation. Prepare a presentation, document, or other medium with which to present this to the CTO.

Module 2

Integration Architecture Styles and Patterns

Reflections

What is your opinion of SOA? Does the explanation in Module 2 line up with your experience?

Is Mule an ESB, in your opinion? Is there a place for an ESB in your organization?

Are microservices realistic? What could prevent them from working for you?

Do design patterns help you or your team design solutions more effectively? Could they?

What architectural styles or patterns would you like to learn more about?

In your organization, what IT infrastructure products support integration? Do these products suggest a certain integration style?

Exercise 2

Architectural Blueprint: Apply Patterns and Styles

After congratulating one another for your impressive presentation to the CTO, you return to your desks to find another meeting invitation in your inbox. Apparently, the architecture team is presenting an introduction to Acme Bank's integration architecture patterns and practices to the entire IT department!

The message went on to stress how important it is to quickly establish and communicate the architecture in order to support the new, aggressive roadmap the CTO is announcing.

Apply the architectural styles and patterns from Module 2, as well as your own experience, to meet the architectural goals you've established and support the CTO's initiatives.

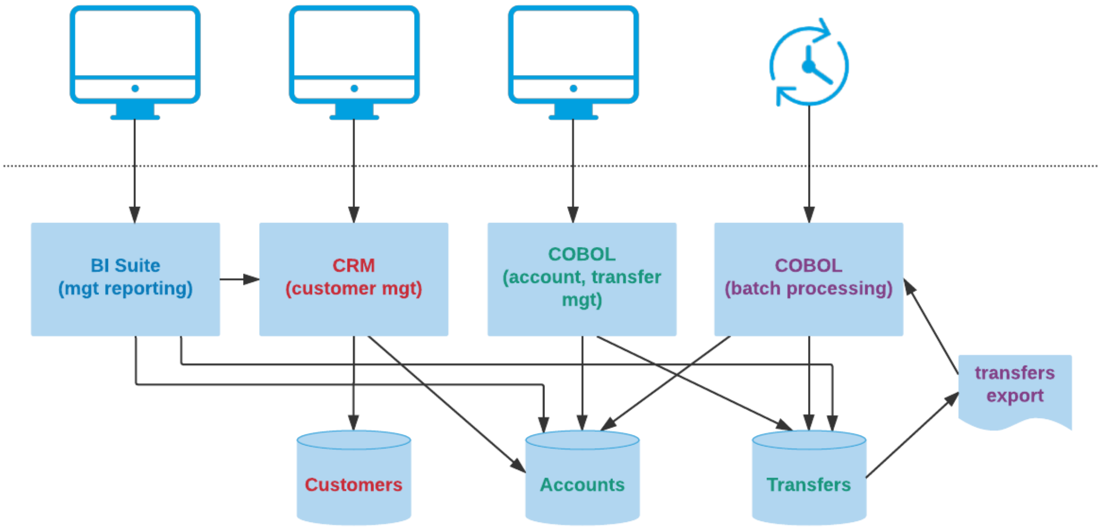
Objectives

Your task is to draft a high-level architectural blueprint for the digital transformation platform. Make sure the following requirements are met:

- Integration of all mentioned systems
- Legacy modernization of the existing (and soon to be replaced) COBOL applications.
- Exposure of main business functionalities through standardized interfaces.
- Offloading the old database so more requests for data can be served, knowing that the existing database has reached its performance limitations.

Technical Background

Partial overview of Acme's applications and systems:



- A mainframe hosts a COBOL application that provides account and transfer management functionalities. Bank employees access the account management application using a terminal. An IBM DB/2 database is used for storing the transfer and account data. Scope is marked in green.
- Another COBOL application for transfer processing functionalities. Scope is marked in purple.
- A CRM system from a commercial vendor with customizations for customer management. Bank employees use a web based UI on top of this Java based CRM system for accessing and managing customer data. The underlying database is an Oracle database. Scope is marked in red.
- A commercial business intelligence (BI) suite for providing management reporting, insight in transfers and customer data. The BI suite connects directly to the DB/2 databases and CRM system. Scope is marked in blue.
- The COBOL applications are difficult to use in an integrated environment as they do not provide proper interfacing methods.
- ACME wishes to have better interfacing capabilities for these applications. In due time, the COBOL applications will be phased out and replaced by a new, yet to be developed Java based application

Recommendations

- Make sure your initial blueprint is future proof, enabling the rapid development of new applications and services on top of the platform.
- Stay at a high level. This will be a relatively short meeting with a large audience.
- Use the MuleSoft approach and recommended architectural concepts in your design.
- Use whatever media your group deems effective. Consider that you'll likely want to keep your blueprint for future exercises and for reference after class.

Module 3

Designing APIs

Reflections

How does interface-first software design influence the software that gets created? What benefits or risks does the approach carry?

In your terminology, what exactly is an API?

Do you think that APIs are constrained to synchronous request-response interactions or do they also support asynchronous communication styles?

Does the MuleSoft categorization of APIs into Experience, Process, and System APIs make sense for your organization? What types of integration might be hard to fit into the model?

What would make a good system API in my environment? What would its data model look like?

When moving to API-led connectivity, what should an organization do about existing integrations and APIs that do not fit the architecture?

What aspects of RESTful web service design seem superfluous, academic, or “ivory tower” to you? Why?

Exercise 3

Architectural Blueprint: Reference API Design

The week after the presentation to the IT department was chaotic. Because of production outages, deployments, and communication hiccups, it took six different meetings to get in front of all the different groups. Although the response was not unanimous, there were several groups who showed enthusiasm about the new direction.

One development team immediately began a rewrite of their current API using REST, ensuring you that they are already experts. Another promised to send you a link to the wiki where they've got these standards "all documented already."

The product manager of a new team formed to quickly deliver a strategic mobile application met with you this morning to express concern about the ability of his team to adopt the architecture in a consistent way with the rest of the IT organization. In the meeting, you agreed to provide a reference API design that would help the team get started doing API-led connectivity in a way that would be consistent with the entire organization.

Create a reference API design that promotes consistent adoption of API-led connectivity. Illustrate how services are decomposed into Experience, Process, and System APIs. Also, include a reference API to illustrate the important aspects of a well-designed RESTful API.

Objectives

Illustrate how to separate responsibilities into Experience, Process, and System APIs.

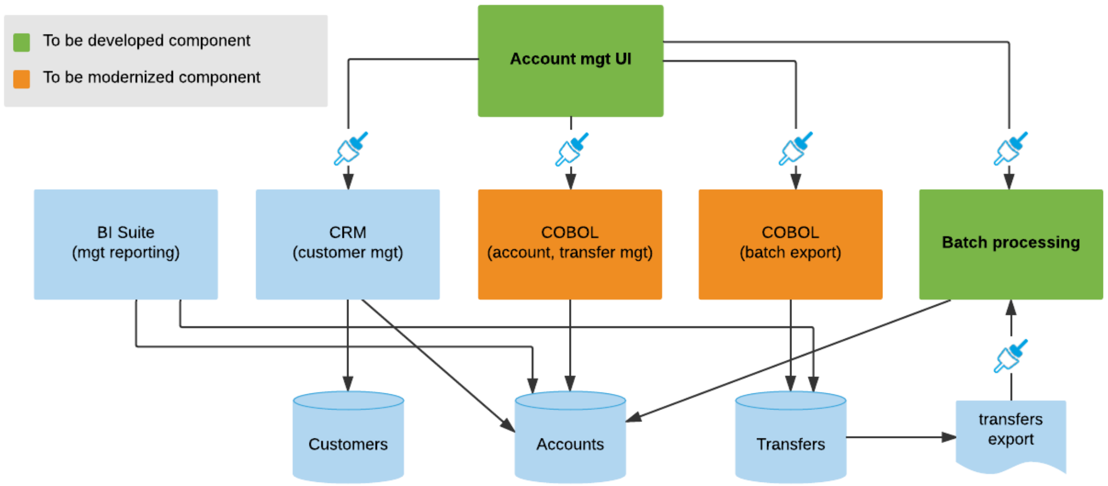
Follow the API design process described in Module 3 to design one of the APIs in your model.

Extra Credit
Create an actual API definition in the Anypoint API Designer, API Workbench or any other text editor of choice.

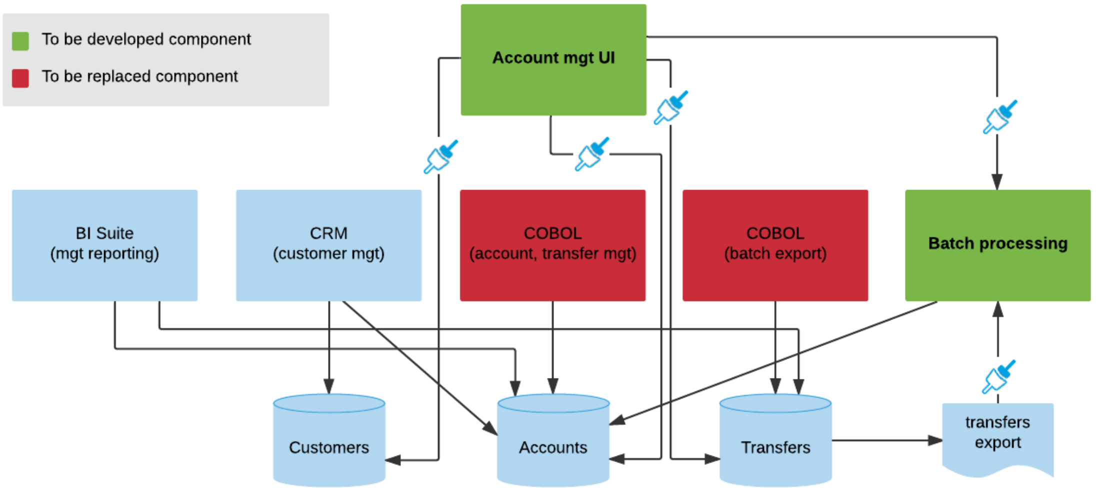
Functional Requirements for Reference Case

Members of your team recently spent time working with the account management application team, so it might serve as a useful basis for the reference case. Refer to the application diagrams below to understand the relevant systems and how they relate to one another.

Phase 1: Legacy modernization



Phase 2: Legacy replacement



User stories completed in recent sprints for the account management desktop application team:

As a bank employee, I want to:
[FR-001] Update bank account information
[FR-002] Replace a customer's information
[FR-003] Retrieve details on a particular incoming transfer
[FR-004] Retrieve a list of all transfers, both incoming and outgoing.

[FR-005] Retrieve a full list of customers
[FR-006] Add an account to a customer
[FR-007] Remove an account from a customer
[FR-008] Retrieve a list of a customer’s outgoing transfers
[FR-009] Create an outgoing transfer from a customer’s account to another account
[FR-010] Add a new customer
[FR-011] Retrieve a detailed account report
[FR-012] Get a list of accounts belonging to a particular customer
[FR-013] Retrieve a list of incoming transfers for a particular bank account
[FR-014] Retrieve details on a particular outgoing transfer
[FR-015] Retrieve details on a particular customer

Recommendations

The RESTful API design process in Module 3:

1. Identify resources
    a. Start small, depth before breadth
2. Define URIs to address them
    a. Think hierarchically, resources can be composed of smaller ones.
3. Determine what consumers need to do with each resource
    a. These will be the HTTP methods.
    b. Think about the differences between similar verbs
        i. retrieve and search
        ii. update and replace
        iii. append and overwrite
4. Design and document resource representation
    a. Choose a syntax (JSON, XML?)
    b. Be consistent
    c. Include examples, schemas if you have time
5. Model relationships between resources with hyperlinks
    a. Include these in the representations where people will find them through experimentation
6. Define response codes
    a. Demonstrate the important distinctions here
7. Decompose into Experience, Process, and System APIs.

For the API definition, use the identified resources and operations as a starting point and add the following elements:

- Documentation and API descriptions
- Media types
- HTTP response types & Sample responses

Module 4

Planning Integration Solutions

Reflections

“Quality” can sometimes mean different things to different people. What aspects of quality seem outside the purview of software architects? What parts of architecture aren’t really covered by quality?

What part does authority play in software architecture, in your experience? What part should it play?

In your opinion, what role should documentation play in software architecture?

How can you determine effective non-functional requirements in your organization?

What causes the greatest inaccuracy in demand and capacity planning? What can you do about it?

When and how is demand planning or performance tuning done in your environment?

Exercise 4

Planning integration architecture

About two months into the digital transformation journey, things seem to be going well. Two internal teams are making unprecedented progress, and they credit the architecture as a major contributor.

Because of an unforeseen market opportunity, Acme Bank has formed a task group to support the Open Bank initiative, and the new Vice President of Software Architecture (your old boss) has asked your team to work with them directly to make sure they see the same success as the more seasoned teams. The task group includes a contract development team that has been assembled to implement all necessary infrastructure and application components. A portion of the team works in Hawaii, including developers, testers, and financial analysts.

You have a product manager on your team who is familiar with the strategic opportunity and is ultimately responsible for the success of the initiative.

Design, in detail, the integration platform components that will allow access to the bank's data coming from various sources. The data must be made available in a uniform, easy to consume format.

Objectives

Determine the most important architectural priorities for the project, and your plan to make sure this initiative is successful.

Write a Software Architecture Document (SAD) that contains all components and details you consider relevant. Feel free to use any documenting/diagramming style or template you prefer.

Recommendations

Make sure the platform is capable of serving multiple different clients, such as mobile apps, web apps, reporting systems, custom clients etc. Keep in consideration that the Open Bank program assumes the availability of public APIs.

Design the platform as an enterprise grade solution. Specify in detail your architectural approach and implementation components. Make sure to pay attention to:

- Architectural viewpoints
- Integration endpoints
- Interface specifications
  - how can the clients consume data from the bank's systems?)
- Data formats and protocols
  - how is the data presented?
- Application layers and components (what are the implementation details?)

You are free to use your own SAD/architectural documents as template. For your reference, check slides 11-17 of module 4. Those slides can help you identifying the most important components of your SAD.

Module 5

Mule Application Architecture

Reflections

What benefits does a template application offer to a Mule integration team? To several teams within an organization?

Do you agree with the statement: "Organizational standards inhibit team productivity and creativity."? Why? Or why not?

Why would an experienced team of Java developers choose built-in Mule components over java custom components?

What aspects of Mule application design are most important to keep consistent across the organization? Why?

What considerations are relevant when deciding between runtime reuse and design-time reuse?

How do automated tests contribute to continuous delivery of Mule applications?

What information should applications expose in a standard way for operations?

Exercise 5

Mule application architecture

After six months of digital transformation, the teams that embraced the new architecture have maintained both the quality improvements and the pace they reported at the beginning. They've begun actively contributing to architecture, and taking the initiative to collaborate with other groups in IT.

A recent surprise was that the team that maintains the COBOL software has recently begun writing the Mule System APIs planned as part of the legacy modernization initiative. They are having some success in dev and QA, and are asking your team for some guidance as to how to build "Acme Quality" Mule applications.

This request echoes similar needs on other teams that are ramping up with Mule. It appears there is a lot of interest in some guidance that is a lot more prescriptive than what your team's been creating.

Objectives

Provide some specific, technical guidance that helps enable teams to build Mule applications that consistently align with Acme's integration architecture.

Build a template application that demonstrates the important aspects of Mule application design at Acme Bank. Include a companion document to help the team understand the relevant aspects of the template application, and the rationale for using them.

Extra Credit

Provide a template for a more specialized application type, and illustrate design decisions that help make that type of application successful and maintainable. Some ideas:

- Bidirectional synchronization process API applications
- System APIs for the COBOL applications
- Experience APIs for mobile consumers