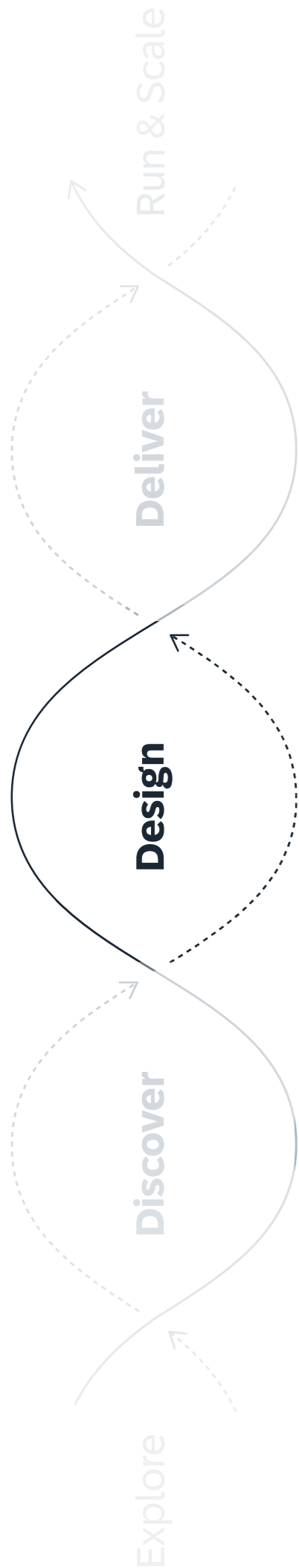# Devise a Conceptual Data Diagram

Describe information objects processed by the architecture.

Why to
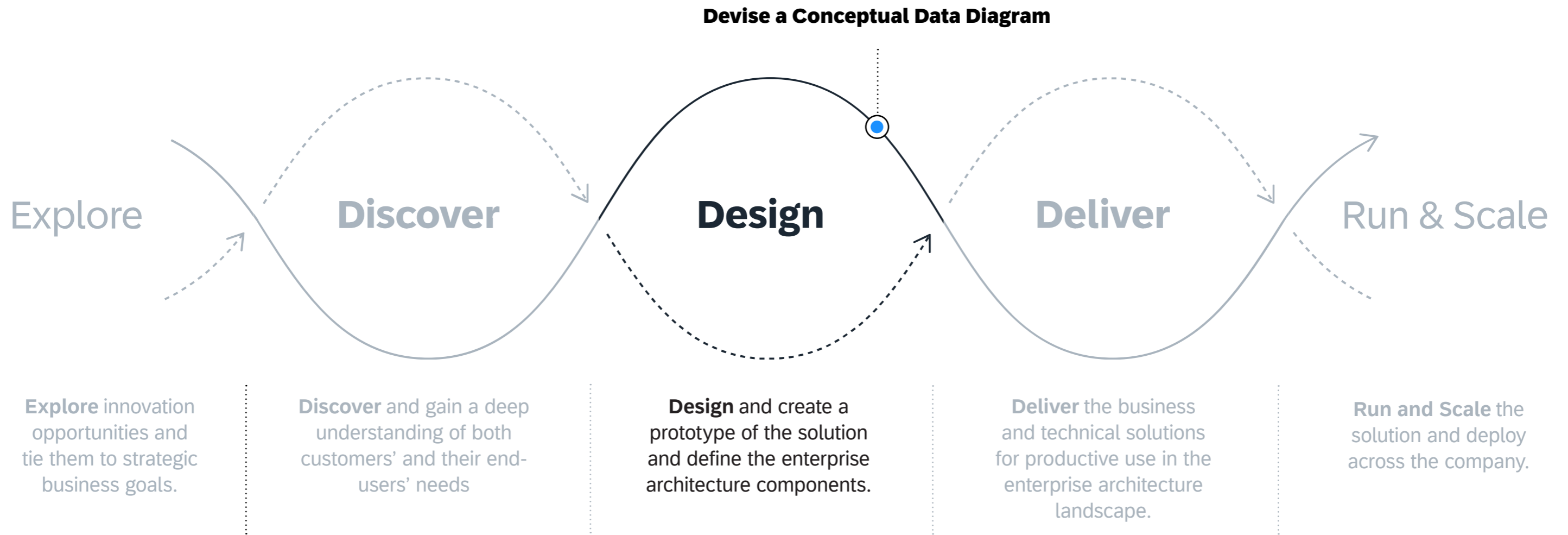# Devise a Conceptual Data Diagram

The purpose of the "Conceptual Data Diagram" is to outline the relationship between data and business entities of the aspired solution.

The Conceptual Data Diagram can be treated like an entity relationship diagram, outlining and describing the information being processed by the aspired solution.

When To
Devise a Conceptual Data Diagram

The Conceptual Data Diagram is created during the Design phase after the
Architectural Decisions List.

Devise a Conceptual Data Diagram

Explore · **Discover** · **Design** · **Deliver** · Run & Scale

Explore innovation opportunities and tie them to strategic business goals.

Discover and gain a deep understanding of both customers' and their end-users' needs

Design and create a prototype of the solution and define the enterprise architecture components.

Deliver the business and technical solutions for productive use in the enterprise architecture landscape.

Run and Scale the solution and deploy across the company.

Learn more about SAP's Human-Centered Approach to Innovation: **https://apphaus.sap.com/approach**

Created by **SAP AppHaus**

member of the SAP
AppHausNetwork

How To
Devise a Conceptual Data Diagram

Duration
60 - 120 Minutes

Participation
1 - 3 People

Roles
Architects

Use Input From
- **Create a Use Case Blueprint Diagram**
- **Craft a Solution Realization Diagram**

Steps

1 Identify information objects (entities) being processed by the solution building blocks (SBBs) in the **Solution Realization Diagram** and understand the information flow between the SBBs.

2 Assign each identified entity a clear, unambiguous name, and add attributes or properties with corresponding data types to provide further detail. For choosing data types, keep it simple and chose types like "number", "string", "date", for example.

3 Think about the relationship between the different entities. Mostly, you will design an association between two entities and define a multiplicity. The association is represented by a solid line between two entities and is described with a verb.

Expert Tips

This diagram is not a technically detailed data model that can be directly mapped to a relational database, for example. Instead, the information objects or data entities represented here may be stored and managed by various solution building blocks within the architecture.

They might not necessarily be stored in a relational database; a graph store might be more suitable, or you might receive the information objects via an API or message queue from a building block outside the scope of your architecture.

Sources of input for creating the Conceptual Data Diagram are the **Use Case Blueprint Diagram** and the **Solution Realization Diagram**.

Example
Conceptual Data Diagram

Business Goal

ID: Integer
Name: String
Description: String
Measure: String

Purpose

ID: Integer
Name: String
Description: String

Project

ID: Integer
Name: String
Description: String
Start: Date
End: Date
Budget: Float
Status: String

0..* *has* 1..*

1..* *supports* 1

0..*

0..*

1..* *requests*

reviews

approves

1 *has* 1..*

Milestone

ID: Integer
Name: String
Description: String
Date: Date

1

Requestor

ID: Integer
Name: String
Department: String

1..*

Reviewer

ID: Integer
Name: String
Department: String

1..*

Approver

ID: Integer
Name: String
Department: String