



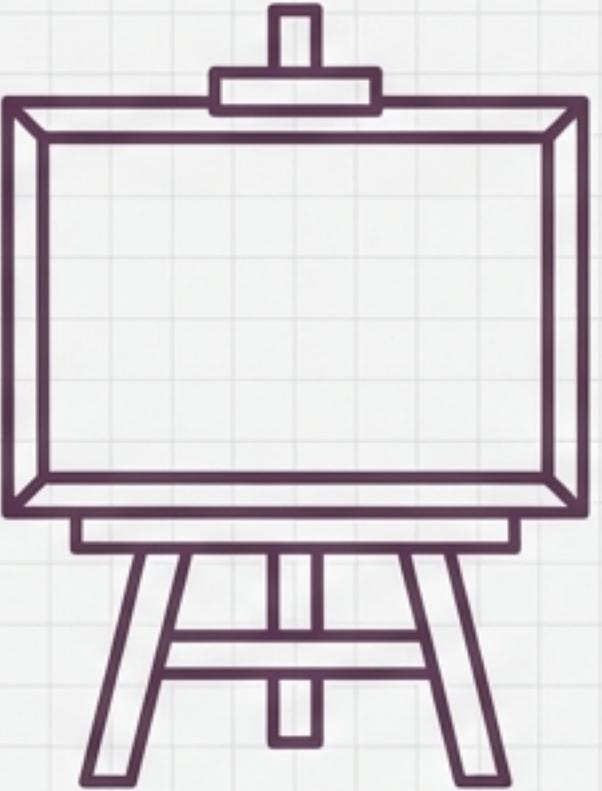
Model-Driven Apps: From Blueprint to Build

A structured guide to designing and creating
powerful business applications.



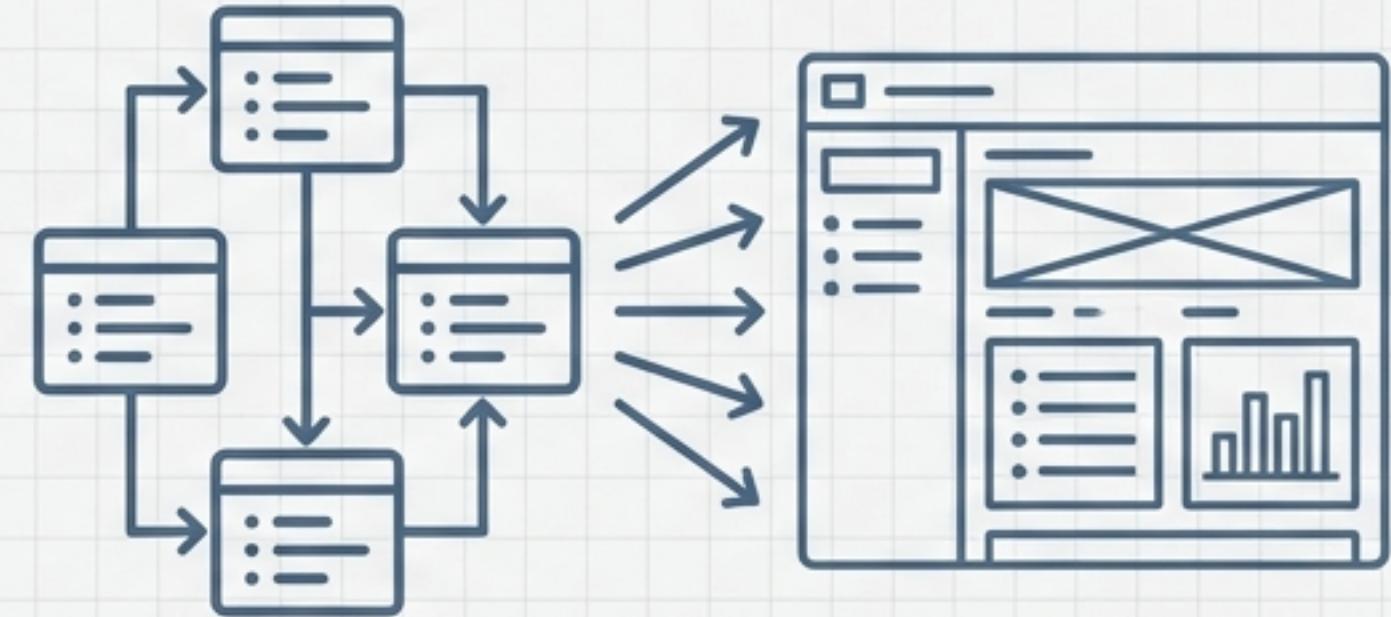
Your Data Model is the Blueprint

Model-driven app design is an approach that starts with your data. The components you add—dashboards, forms, views, and charts—determine the layout. This allows you to focus on quickly viewing business data and making decisions, rather than on intricate app design.



Canvas Apps

The app maker has total control over the app layout and user experience.

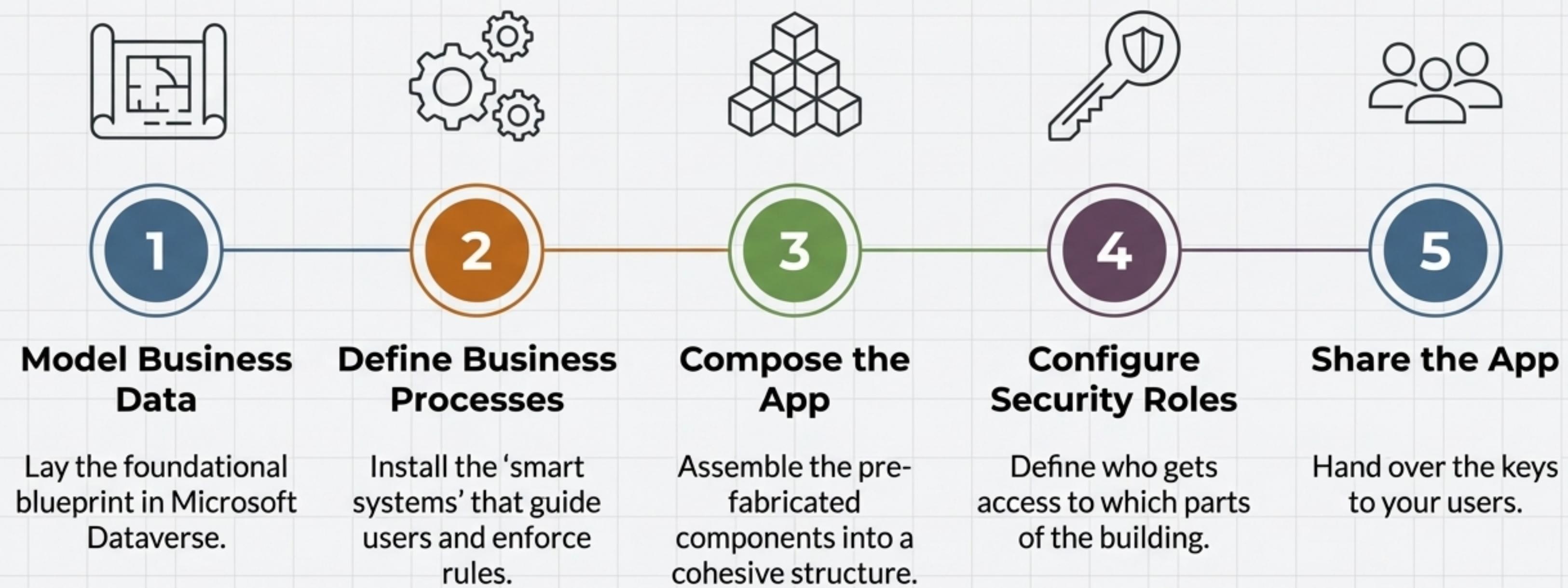


Model-Driven Apps

The layout is largely generated for you based on the components you add to the underlying data model.

The Five Phases of Application Construction

We will follow a five-step process that takes us from initial concept to a fully functional, shared application.



Step 1: Lay the Foundation with Your Data Model

The data model is the heart of your app. It defines what data you need and how it relates to other data. This structure is built in Dataverse using four key components.

Tables

The primary containers for your data, like an `Account` or `Contact` table.

Columns

The properties you track within a table, such as `Email Address` or `Contract Amount`. Each has a specific data type (text, number, currency, etc.).

Relationships

Definitions of how tables connect (1:N, N:1, N:N). A lookup column creates a 1:N relationship.

Choice Columns

A special column type that gives users a predefined set of options to choose from.

Pet Grooming

Pet Table

Pet Name (Text)

Appointment Date (Date/Time)

Species (Lookup) -----;

1:N Relationship

Species Table

Species Name (Text)

Dietary Needs (Text)

Step 2: Construct the User Interface

UI components determine how users interact with the data you just modeled. Power Apps generates most of this for you based on your selections.

Forms

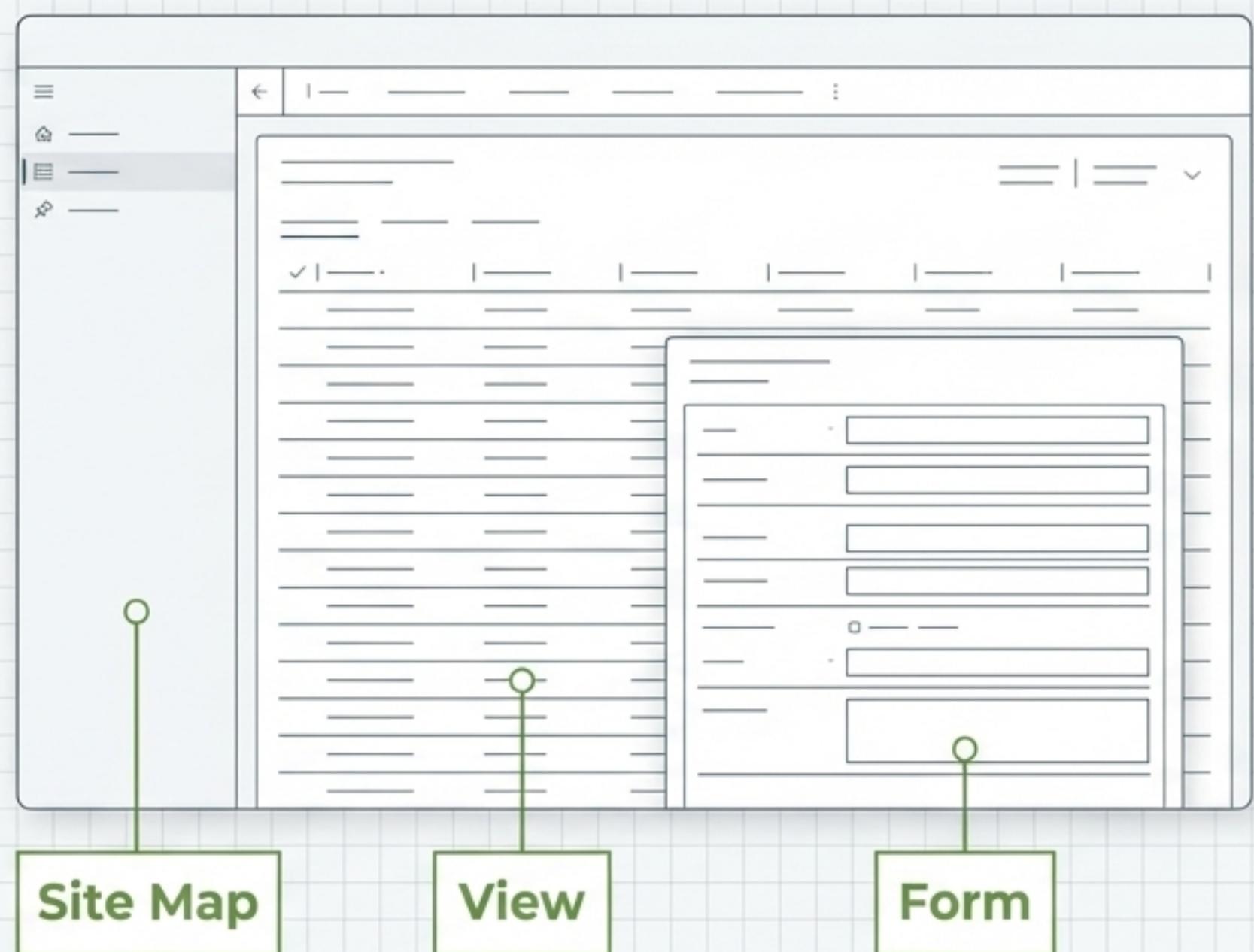
A set of data-entry columns for a given table. This is where users create and edit individual records (e.g., a 'New Prospect' form).

Views

Define how a list of records for a table is displayed. A view determines the columns shown, their width, sort order, and default filters.

Site Map

Specifies the navigation for your app, helping users move between different views and dashboards.



Step 3: Install the ‘Smart Systems’ with Business Logic

Defining and enforcing consistent business processes is key. This ensures users follow the same steps every time, reducing training and preventing errors.

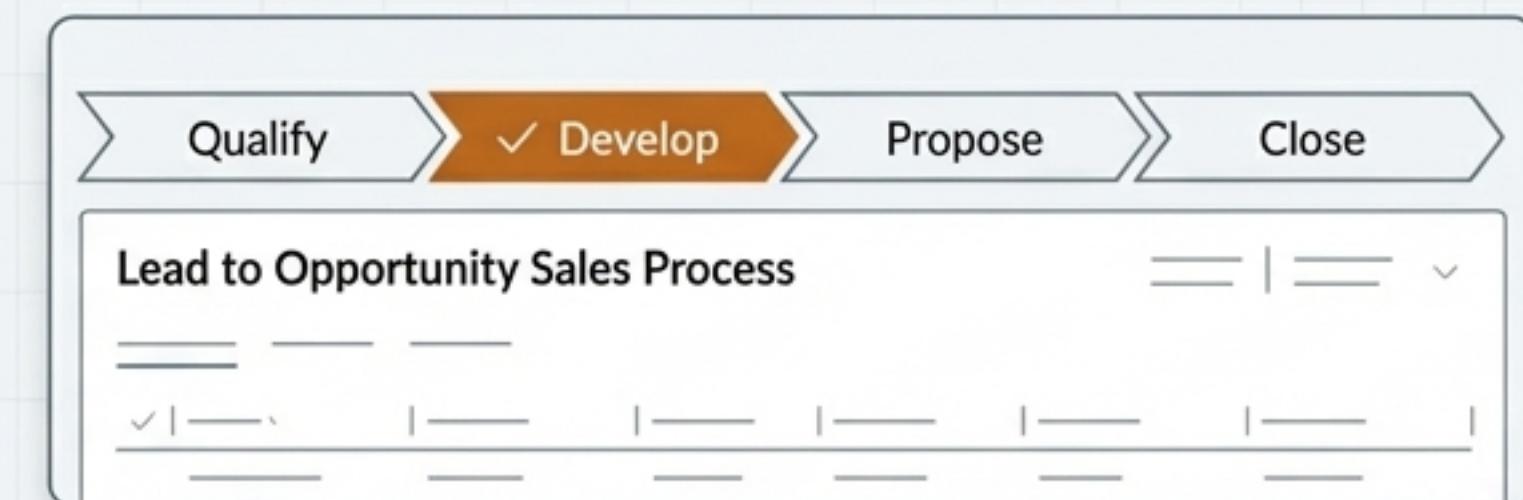
Business Rules

Apply rule-based logic to a form without code. Use them to set column requirements, requirements, show/hide columns, or validate data.

The image shows a user interface element consisting of two parts. The top part is a dropdown menu labeled "Travel Type" with the option "Automobile" selected. The bottom part is an input field labeled "Mileage *". An orange arrow points from the text "Business Rules" to the "Mileage *" field, indicating that business rules can be applied to validate data in specific fields.

Business Process Flows (BPF)

Visually guide users step-by-step through a process. They can span multiple tables (up to five) and lead users to a desired outcome.



*Example: In an expense tracking table, the ‘Mileage’ column is required *only if* the ‘Travel Type’ is ‘Automobile’.

*Example: The ‘Lead to Opportunity Sales Process’ guides a seller from initial lead, to quote, to order, to invoice.

A Closer Look at Logic Components

Type of Logic	Description	Designer
Business Process Flow	An online process that walks users step-by-step through a standard business process.	Business process flow designer
Workflow	Automates business processes within Dataverse. Can be triggered manually or automatically based on events like record creation.	Workflow designer
Business Rule	Applies form logic to set column requirements, hide columns, or validate data using a simple interface.	Business rule designer
Power Automate Flow	A cloud-based service to create automated workflows between apps and services (e.g., get notifications, sync files). Reaches beyond Dataverse to other applications.	Power Automate

Key Takeaway

Use Business Rules for form-level validation and consistency. Use Business Process Flows to create guided, multi-stage user journeys.

Step 4: Visualize Your Data for Actionable Insights

Your app needs to provide high-level snapshots of business data. Visualizations allow users to understand trends and drill down for more detail.

Charts

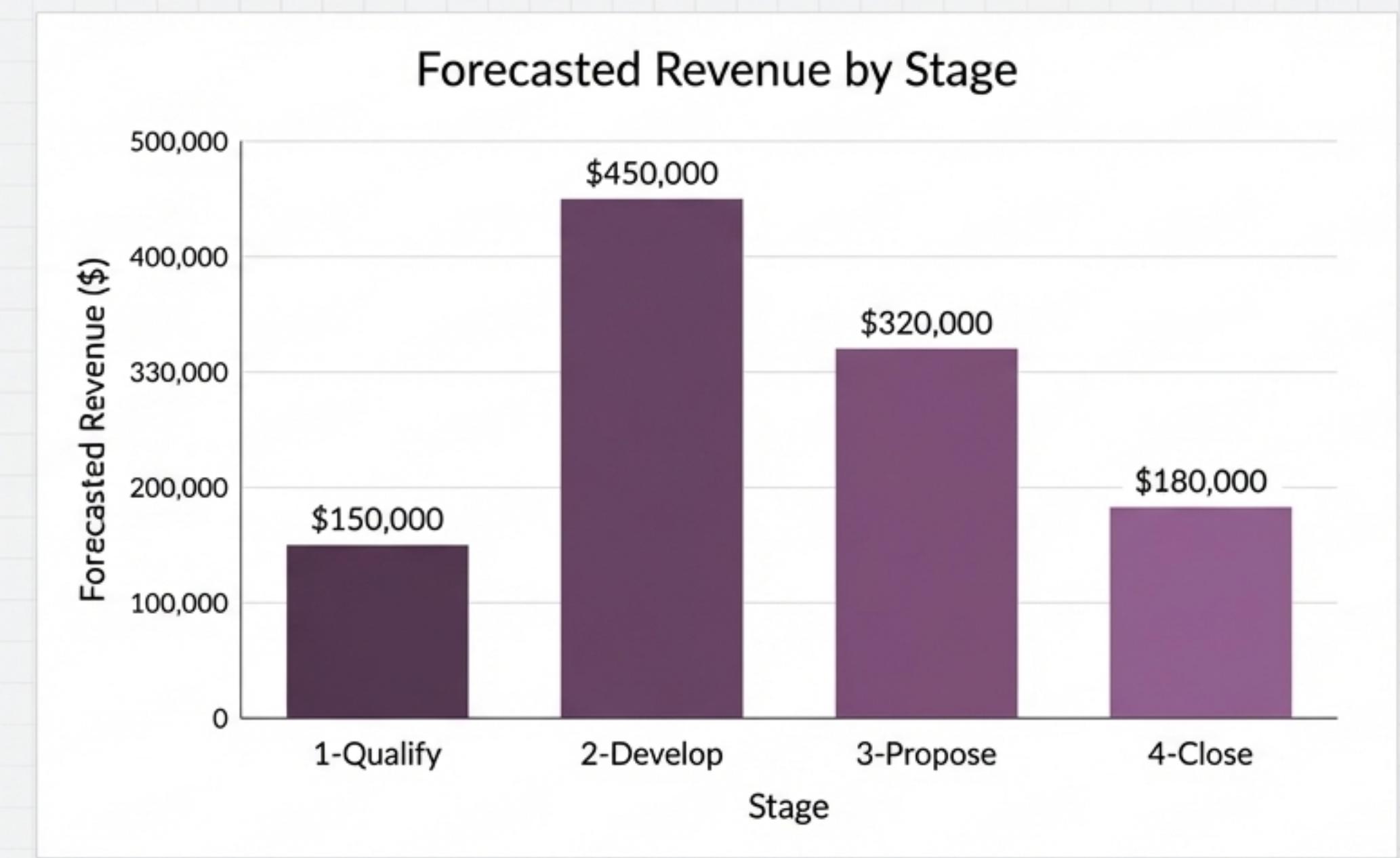
A single graphic visualization (bar, pie, line) that can be displayed within a view, on a form, or on a dashboard.

Dashboards

A palette for one or more visualizations that provides an overview of actionable business data.

Embedded Power BI

Power BI tiles and dashboards can be added for advanced business intelligence insights.

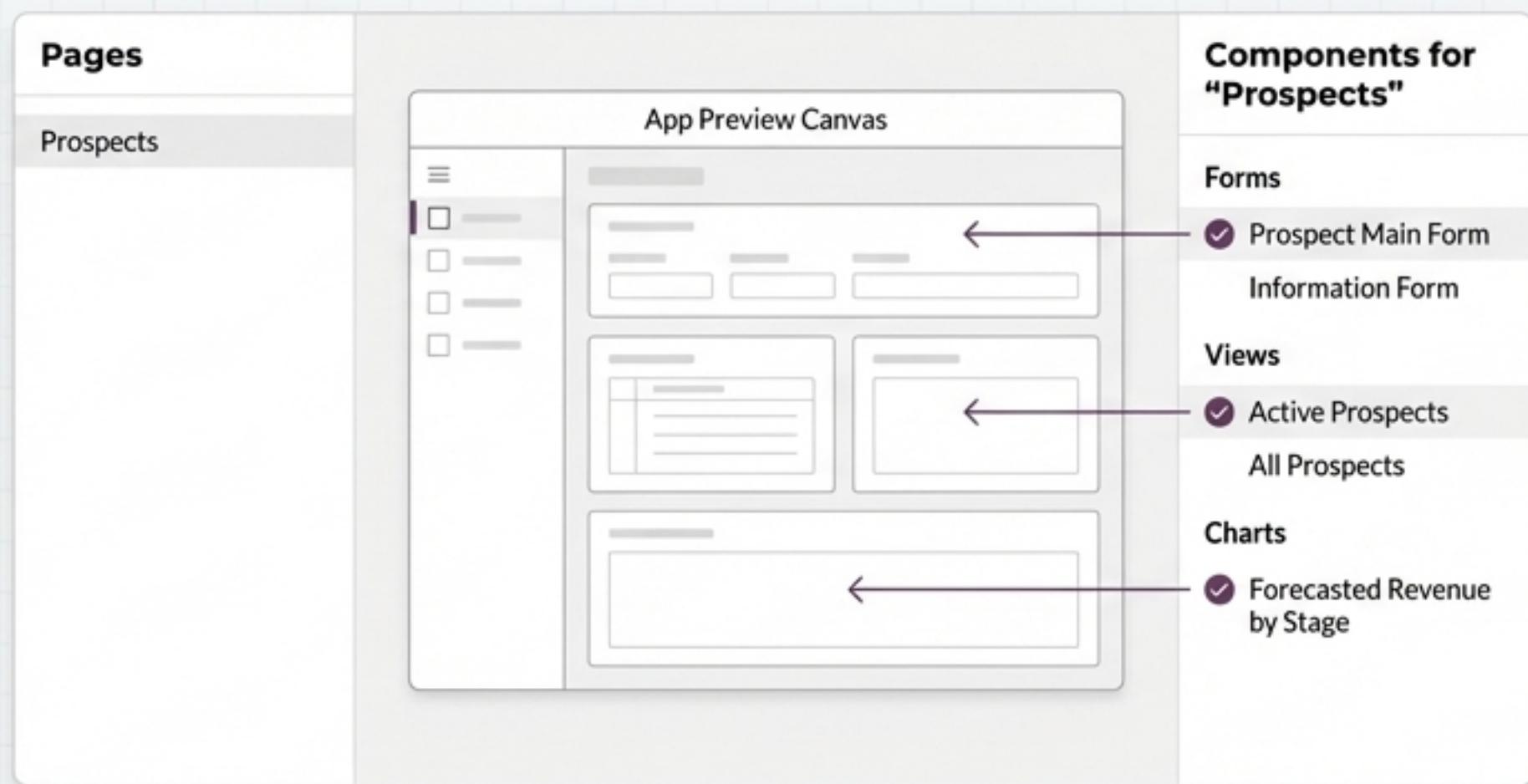


Composing the App: Assembling the Pieces

After modeling data and defining processes, you build your app in the App Designer.

- 1. Start with a page design:** Choose a layout, such as a blank page with navigation.
- 2. Add pages:** Add pages based on your Dataverse tables (e.g., add the 'Prospects' table).
- 3. Select Components:** The designer allows you to include the specific Forms, Views, and Charts you've created for that table.
- 4. Publish:** One click saves your changes and makes the app available.

Key Insight: Power Apps automatically creates a **site map** for you, which defines the app's navigation based on the pages and components you add.



Step 5: Configure Security Roles to Control Access

Access to a Dataverse table requires an assigned security role. This is how you control who can access restricted or sensitive data in your app. Every app user must have at least one security role.

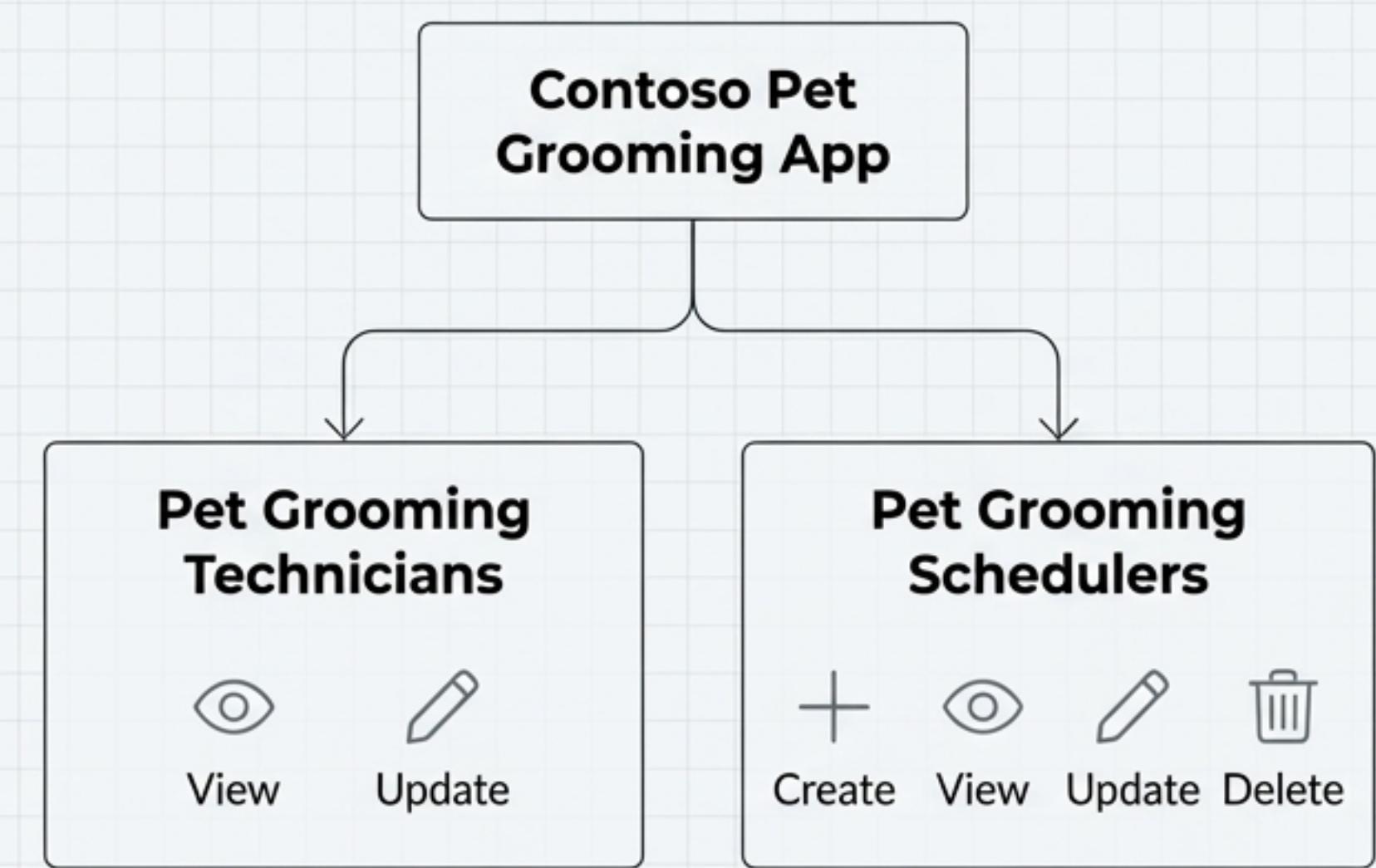
Scenario: Contoso Pet Grooming App

To manage privileges for the custom Pet table, we must create custom security roles.

- **Pet Grooming Technicians:** Need to view and update appointments.  
- **Pet Grooming Schedulers:** Need full control to create, manage, and delete appointments.    

Key Point

Predefined security roles exist, but they only apply to standard Dataverse tables. For custom tables, you must create your own.



A Granular Look at Security Privileges

Privilege	Description
Create	Required to make a new record.
Read	Required to open and view an existing record.
Write	Required to make changes to a record.
Delete	Required to permanently remove a record.
Append	Required to associate the current record with another (e.g., attach a note to an opportunity).
Append To	Required to associate a record with the current record (e.g., add a note to an opportunity).
Assign	Required to give ownership of a record to another user.
Share	Required to give access to a record to another user while keeping your own access.

Example Application

Technicians Role: On the Pet table, they have **Read, Write, Append** privileges set to 'Organization'.

Schedulers Role: On the Pet table, they have **Create, Read, Write, Delete, Append, Append To, Assign, Share** privileges set to 'Organization'.

The Final Step: Sharing Your App

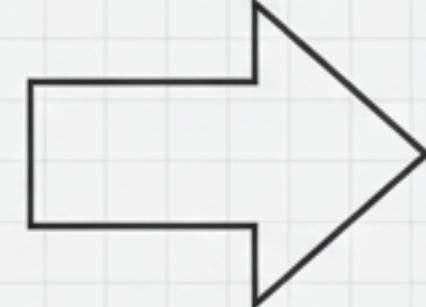
Sharing a model-driven app is a two-step process to ensure both the app and its underlying data are secure.

1



Step 1: Assign Security Roles to Users

In the Power Platform admin center, find the security role you created (e.g., 'Pet Grooming Technicians'). Add the specific users or teams who should have that role. This grants them permission to access the data in the tables.



2



Step 2: Share the App with Users

From the Power Apps maker portal, select your app and click 'Share'. Share the app with the same users or teams you assigned roles to in Step 1.

Critical Takeaway

You must complete both steps. If you share an app with a user who does not have an assigned security role for the tables the app uses, they will not be able to open or use the app.

The Finished Build: A Process-Driven Application

Summary of the Journey: By following a structured, data-first approach, you can efficiently build powerful business applications.

- You started with a solid **data blueprint**.
- You constructed an intuitive **user interface**.
- You installed intelligent **business logic**.
- You created insightful **visualizations**.
- You secured it with granular **security roles**.

The Result: A robust, secure, and consistent application built with a focus on business process and data, not complex, custom-coded design.

