Introducing UltraFlow

by MIRA Innovations

Safe Harbour Statement

UltraFlow is in the final stages of development.

The features, functionality and appearance of the final product may differ from that described and illustrated in this document.

Business decisions should be made based on current, released products.

MIRA Innovations accepts no responsibility for decision made on the basis of this of any other draft document describing UltraFlow or any of its other products.

The Demand for Omni-Channel Service

Increasing numbers of citizens and consumers expect to be able to access information, services and goods online, 24/7, from any device.

They also expect service providers and businesses to know who they are, what services and products they already have, and to guide them to other offerings they may be interested in.

This is especially true when citizens interact with public services, where they expect to be able to access all providers via one common interface.

Meeting these expectations is challenging:
- many organisations still rely on paper-based communication with citizens and consumers
- legacy systems are often hard to access
- some technologies are not accessible by all channels
- regulatory and legal changes result in constant change to processes

Companies such as Salesforce have excellent offerings that can solve many of these challenges, However, there are still significant obstacles that have to be solved:
- customisation is almost always required
- configuration (clicks, not code) has limits
- custom code is usually
  - expensive to create
  - expensive to test
  - expensive and difficult to maintain

The global challenges of the pandemic and extreme weather events have highlighted the urgent need to transition to digital interactions with citizens and consumers.

How can this be achieved quickly, and within limited budgets?

Meet UltraFlow

UltraFlow is a product that is available on the Salesforce AppExchange.
It helps Salesforce clients and vendors configure, build, test and deploy solutions that involve complex processes and / or many user-facing forms, with the bare minimum of custom code.

There are three core principles.

Simplicity

UltraFlow Designer is easy to use.
The four tools are intuitive, with drag and drop functionality for building forms and flows.

Flows can initiate other flows and sub-processes, allowing complex processes to be broken down into more manageable components.

Reusability

Data Fields represent your data points, and are decoupled from Salesforce objects. Use them in user-facing forms.

Forms are decoupled from flows. Configure a form to collect a name and address and use it in any flow where that information is collected by dragging and dropping it into the flow.

Tasks are also decoupled from flows. Tasks describe Apex methods that perform server-side processing for you. These methods implement UltraFlow's unique interface, which allows them to be written as object-oriented code - simple to write, simple to test, simple to maintain, and eminently reusable.

Consistency

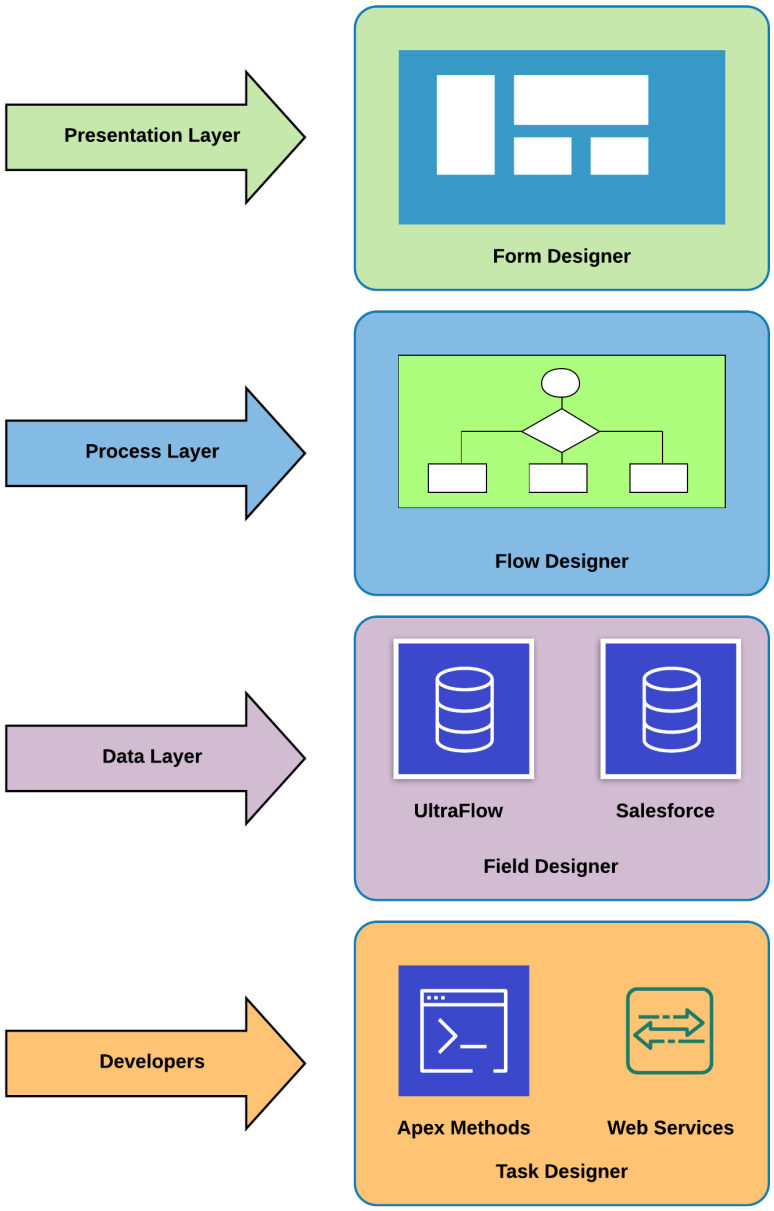
At runtime, forms are rendered by UltraFlow's unique framework - no front-end code is required. This framework ensures that all of the user-facing forms have the same look and feel.

There are options to modify the style - for example, if a form is part of an application for car insurance, configure the background to show a picture of a car when collecting an applicants details.
This style is applied at the **flow** level and inherited by all forms (and sub flows) in the user journey - so the same form that has a picture of a car in the background when applying for car insurance can have a picture of animals when it is used for an application for pet insurance.

UltraFlow Conceptual Architecture

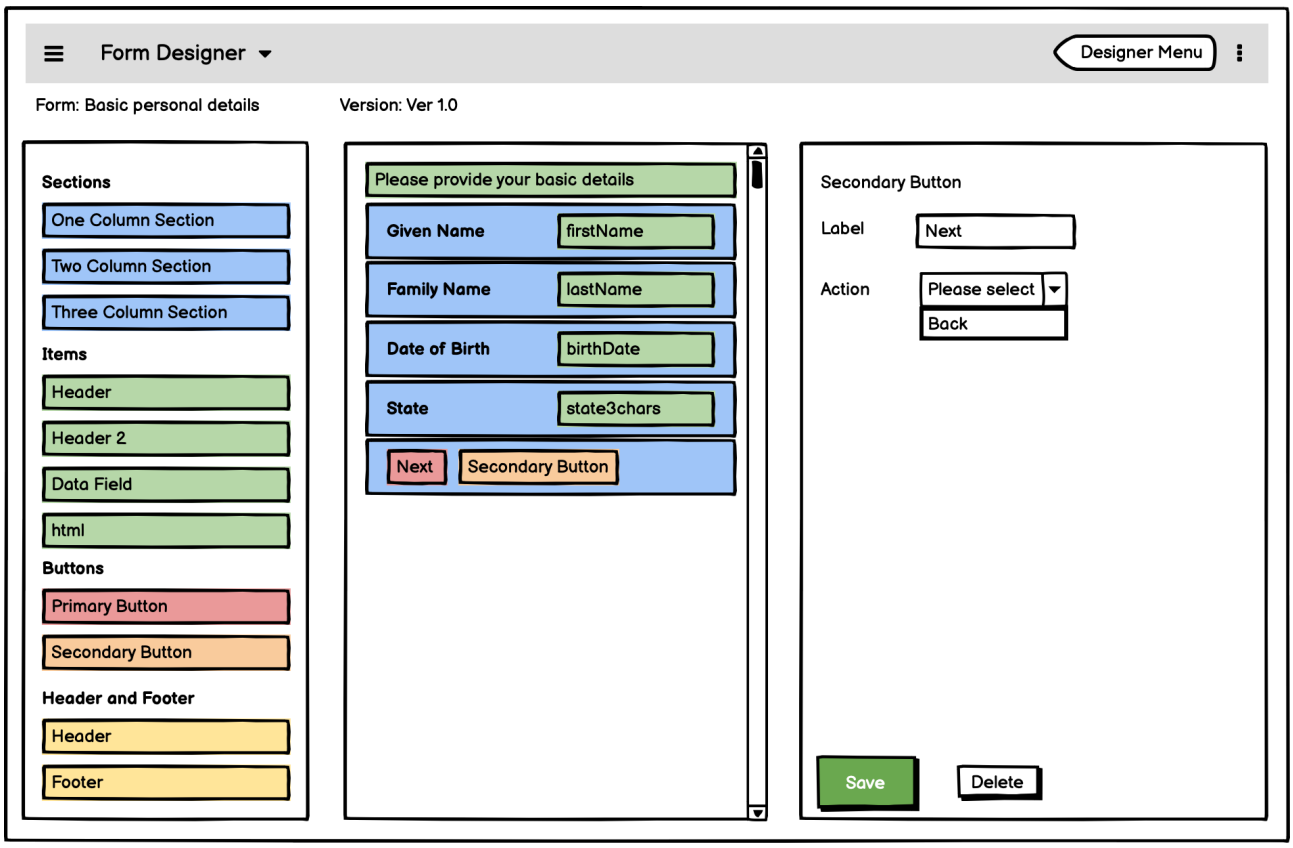
Let's have a closer look at UltraFlow.

There are four conceptual layers:



The Presentation Layer

Forms are configured using the Form Designer.



Drag and drop sections and items from the left-hand panel onto the central canvas, then configure them in the right-hand panel.

Create, save, update and clone Forms with ease.

At runtime, the UltraFlow runtime engine loads and parses a Flow.

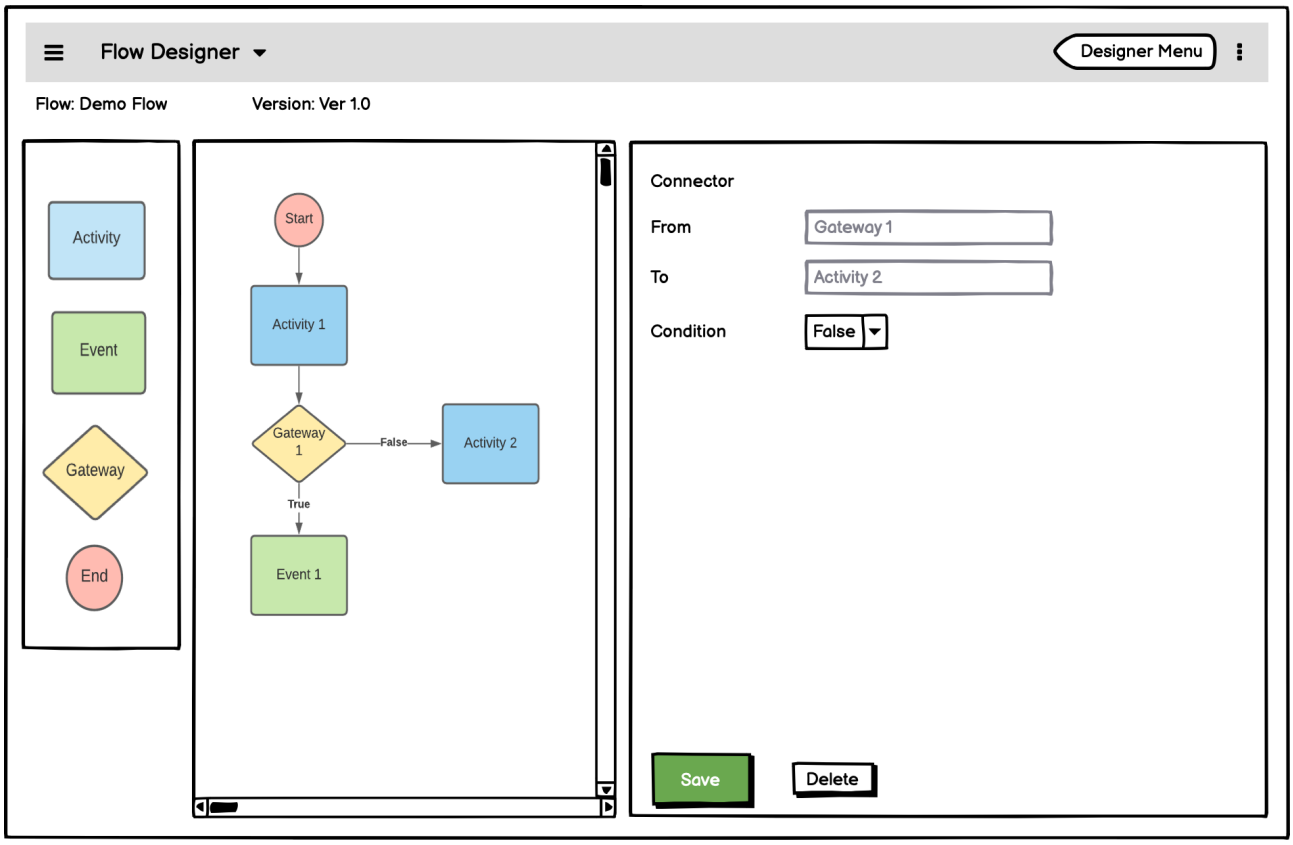
When it reaches an Event element, it retrieves the metadata for the specified Form and renders it and its data in its unique Lightning framework - much like the way a browser renders html.

When the user clicks on a button, this framework passes the data back to the runtime engine for processing.

The payload is very small in both directions, giving lightning fast page rendering.

The Process Layer

Flows are configured using the Flow Designer.



Drag and drop elements from the left-hand panel onto the central canvas, then configure them in the right-hand panel.

Click and drag between elements on the canvas to create connections.

Configure styling for the Form background, headings, sub-headings, labels, field text, buttons and more.

Create, save, update, clone and test Flows with ease.

When a Flow is ready, use UltraFlow Publisher to deploy to your other Salesforce orgs.

At runtime, the UltraFlow runtime engine loads and parses a Flow, finds the start element, then processes each element in turn.

Use navigation events to start a sub-Flow (with Forms) or sub-process (no Forms, only Activities and Gateways).

The runtime engine's breadcrumb trail provides the user with a seamless experience navigating between Flows with 'next' and 'back' buttons.

The Data Layer

There are two parts to the data layer - UltraFlow's master directory, and Salesforce records.

At runtime, the master directory comprises key / value pairs for the Data Fields used on the Forms rendered during a user's journey.
This data is encrypted at rest, and is visible only to the user who began or resumed an UltraFlow journey. Data is discarded when the journey is complete.

Storing runtime data in this way allows UltraFlow to pre-populate Data Fields that appear on multiple Forms, saving the user time.

When required for legal reasons, it is possible to retain this transactional data and make it visible to an administrator who has access to the decryption tools.

Data Fields are configured using the Field Designer.

The data collected in the master directory for most Data Fields will eventually be saved to a Salesforce record by a Task.

The Field Designer has a feature that allows Data Fields to be created from Salesforce objects in bulk.
Simply select the object, choose which fields are required, and click.

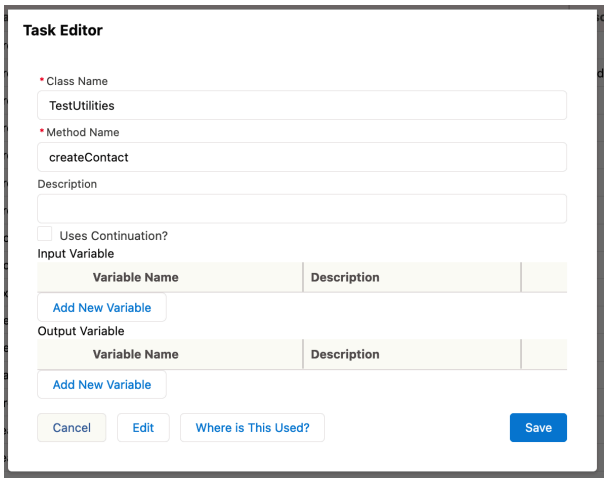
Data Fields are created using the Salesforce field's attributes - label, data type, length, decimals and so on.

Developers

UltraFlow has many unique features that allow developers to create modular, efficient, reusable and easily maintained code.

For example, when a method reads a Salesforce record, it can store it in a buffer where it is available to other methods that execute later in the Flow as the user completes their journey.

Developers are given the requirements for a method when a Task is created using the Task Designer.



A non-developer still knows what needs to be done, what data is needed for the Task, and what results are required.

They configure this information in the Task Designer along with a description of the business requirement.

Developers can see these Task records and use the information to write a method that will complete the Task.

The method has strictly defined input and output variables, and is not required to perform complex processing - that is what the Flow is for.

Methods written for UltraFlow Tasks:
- have no awareness of the context in which they are being executed
- are in a class that implements the UltraFlow taskInterface
- can initially be written to return a hard-coded 'happy path' output so that the Flow can be tested quickly
- when completed, will have very high code coverage from test methods
- are deployed outside of UltraFlow with any CI/CD tools

At runtime, the UltraFlow runtime engine loads and parses a Flow.
When it reaches a Task element, the runtime engine
- loads the Apex class into memory
- uses the configured variable mapping to place data into an input package
- executes the method
- handles any errors
- uses the configured variable mapping to place data into an output package
- if so configured, updates the data values in the master directory

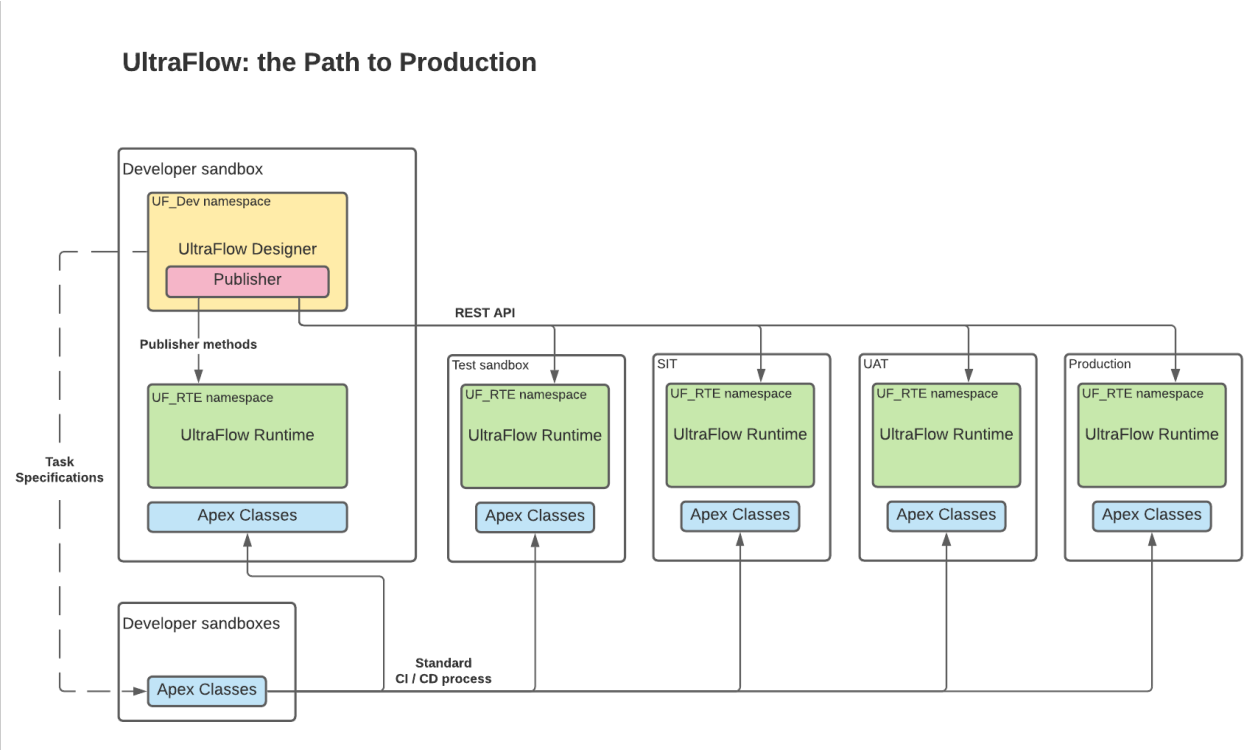
Publishing - the Path to Production

UltraFlow is delivered as two packages: the Designer, and the Runtime Engine.

Install the Runtime Engine in each Salesforce org in your deployment path, except individual developer sandboxes.

Install the Designer in one sandbox only.

This diagram illustrates how Forms and Flows are published:



Fields, Forms and Flows are published in the Designer's sandbox by the Field Designer, Form Designer and Flow Designer tools respectively.

Flows first have a status of ‘In Development’, can be tested directly from the Flow Designer, and can be updated or deleted.

Once published to any other Salesforce org, the status changes to ‘Deployed’, and the Flow cannot be updated or deleted.

When changes are required, clone the Flow in the Designer sandbox to create a new version.

Feature Comparisons

UltraFlow has many features that differentiate it from other flow and process managers.

This table highlights the important differences.

Feature	Salesforce Flow	Vlocity OmniStudio	MIRA UltraFlow
Configurable User Interface	Yes (1)	Yes (2)	Yes
Configure business process without writing code	Yes (3)	Yes (4)	Yes
Supports simple integration	Yes (5)	Yes (6)	Yes (7)
Supports complex integration	No	Yes	Yes
Supports Apex continuation pattern	No	No	Yes
Supports redirect and return (e.g. to a payment gateway)	No	No	Yes
Supports multiple business processes	Yes (8)	No	Yes
Supports subflows	Yes (9)	No	Yes
Intuitive UI builder	Yes	No	Yes
Built-in consistent user experience	No	No	Yes
Flows can be versioned	Yes	?	Yes
Multiple versions can be active at the same time	No	?	Yes
Forms can be versioned	No	?	Yes
Forms can be re-used	No	No	Yes
User can save and resume a session	No	?	Yes
Data is encrypted at rest	n/a	n/a	Yes
Data fields are pre-populated when re-used inside a flow	No	No	Yes
Fully supports multiple languages	No	No	Yes
Licensing	n/a	Per seat	Per org

1 - Screen Flows. Limited options extendable with Lightning Web Components
2 - FlexCards
3 - Flows are always paired with a database transaction
4 - OmniScript
5 - External Services and custom Apex for API calls
6 - Integration Procedure and DataRaptors
7 - Custom Apex code
8 - Flow Orchestrator (Beta)
9 - introduced in Winter 22