Dunnington Software Development Incorporated

Moving At The Pace Of Integrity!

Pega Constellation UI Framework: An Analysis of Views, Forms, and Data Integration

Executive Summary

This document provides a comprehensive synthesis of the Pega Platform's Constellation architecture, focusing on its principles and practical application for user interface (UI) development. The Constellation design system is a prescribed, component-based framework designed to accelerate development, ensure UI consistency, and improve application scalability and maintainability. It achieves this by separating front-end concerns from back-end logic, allowing roles such as Pega Business Architects to configure sophisticated user experiences directly within App Studio.

The core building blocks of the Constellation UI are **Views**, which serve as the reusable link between user interactions and the underlying Case data. The system automatically generates foundational Views—such as the Full Page, Details, and Pulse Views—to establish a consistent structural framework. For data collection, **Forms** are utilized, which can be organized into complex, user-friendly workflows through **Multi-step** or **Hierarchical** structures depending on whether the user journey is sequential or non-sequential.

A pivotal feature for efficiency is the concept of **Primary Fields**. By designating the most critical data elements within a Data Model, these fields are automatically populated across the system's Create, Edit, and Details Views. This mechanism is enhanced by an inheritance pattern where Primary Fields from referenced data objects are automatically displayed in a Case Type's Views, significantly reducing redundant UI configuration and enforcing data relevance across the application.

1. The Pega Constellation Design System

The Pega Constellation architecture represents a fundamental shift in UI development, moving away from traditional section-based rules to a more modular and efficient model. It is built on the **Principle of Separation of Concerns**, which divides an application into distinct parts, each addressing a specific concern, thereby promoting software modularity and flexibility.

1.1. Core Concepts

A **design system** is a systemized collection of design rules, UI elements, and UX patterns that ensures a consistent user experience and speeds up development. The **Constellation design system** is Pega's specific implementation of this concept.

- **Prescribed Design:** Constellation provides a complete, prescribed presentation layer with a set of informed designs and templates. The core front-end experience, including information architecture, interactions, and workflows, is pre-defined but remains configurable.
- **Builder Kit:** It acts as a complete UX toolkit, offering a shared library of UI elements, styles, UX patterns, components, and themes. This is specifically optimized for large organizations with complex Case Management use cases.
- **Configuration:** Application development and design, particularly the configuration of Views, is performed entirely in App Studio by Pega Business Architects (BAs), Citizen Developers, and System Architects (SAs).

1.2. Key Advantages

The adoption of the Constellation design system provides several business and technical benefits:

- **Accelerated Development:** Pre-set configurations and a complete front-end library reduce design and development time, allowing for faster delivery of business value.
- **Consistency and Scalability:** The systemized collection of rules and patterns ensures a consistent user experience across applications and channels, making design work scalable and repeatable.
- **Seamless Updates:** Because the core front-end experience is highly prescribed, application updates for future releases are seamless.
- **Improved Accessibility:** Constellation offers significant improvements in out-of-the-box accessibility compliance, including compatibility with screen readers and localization.

2. Fundamental UI Components: Views and Forms

The user interface in a Constellation application is constructed from a set of core components, primarily Views and Forms, which dictate how users interact with and process Case data.

2.1. Understanding Views

A **View** is a fundamental component of the UI that serves as the bridge between user interaction and the data required to resolve a Case. It controls the specific data a user inputs, selects, or sees during a given Task or Assignment.

- **Functionality:** Views are populated with various UI elements such as text fields, checkboxes, lists, and galleries. The data captured is stored in Pega Platform or an external system, making it reusable.
- **Reusability:** A key design principle for Business Architects is to design Views for reuse. A single View, such as one displaying product details, can be applied across multiple Case Types within an organization.
- **Configuration:** To create a successful View, a Pega BA must consider the user workflow, determine the data needed for each task, and decide on the best way to display that information.

2.2. Types of Views in Constellation

Constellation provides four primary types of Views for capturing and displaying data during Case processing.

View Name	Description	Editability
Full Page View	The main interface for processing Case work or accessing information. A Full Page View is automatically created for each Case or data object.	Read-only
Partial	A View created to display information related to a specific Case or data object.	Read-only
List	A View that displays data from a source as a table, timeline, or tile-based gallery. It can be configured with filters,	Read-only

	sorting, bulk actions, and the ability to edit or add new records.	
Form	A View designed to gather information from users while they work on a Case. Default Form Views include **Create** (to capture data for an Assignment) and **Edit** (to modify important fields).	Editable and Read-only

2.3. Automatically Generated Views

When a new Case Type or data object is created, the Pega Platform automatically generates a set of default Views to provide an immediate visual framework.

- **Full Page View:** This is the primary visual framework for a Case, featuring a three-pane layout optimized to display a large amount of information without excessive scrolling.
- **Details View:** An optional, read-only View configured by default to display the **Primary Fields** of a Case, offering a quick summary of critical data.
- **Pulse View:** An optional View that displays the Pulse feed, enabling users to post, view, and reply to messages for collaboration within the context of the Case.

2.4. In-Depth Look at Forms

While all Forms are Views, not all Views are Forms. A **Form** is a specific type of View designed as an interface for collecting data from users to process work. It is preconfigured with action buttons like Cancel, Next, Previous, and Submit.

- **Configuration:** Forms can be configured on the **Workflow** tab for a specific Step in the Case Life Cycle, or on the **UX** tab of a Case Type or data object.
- **Layout:** Forms can utilize one to three columns. The best practice for most use cases is a two-column layout. The Constellation design system automatically adjusts the size and width of fields based on their type to enhance usability and ensure logical tabbing order (left to right).
- **Instructions:** A form View can display the default instructions configured for a Case Step, or these can be overridden with custom text or hidden entirely.
- **Field Groups:** To organize related data within a single View, a **field group** can be used. This clusters individual fields under a single header and can include instructions or be configured as a collapsible section. Field groups are specific to the View in which they are created and are not reusable.

3. Managing Complex Data Entry

For assignments that require a large amount of data entry, Constellation provides specialized Form structures to guide users and improve the experience.

3.1. Multi-step Forms

A **Multi-step Form** breaks down a single, complex assignment into a guided, linear workflow composed of multiple focused and concise Views.

- **Purpose:** They are ideal for helping users complete complicated tasks that must be performed in a sequential order, such as an online order process.
- **Navigation:** Users navigate between screens using Next and Previous buttons. The form can display a horizontal or vertical navigation menu to indicate the user's progress through the steps. The Submit option appears only on the final screen.

3.2. Hierarchical Forms

Like Multi-step Forms, **Hierarchical Forms** divide a complex assignment into smaller, related groups of fields to reduce scrolling.

- **Structure:** Instead of using multiple Views, a Hierarchical Form splits information across multiple tabs within a single Form View. Each tab is referred to as a **Form group**.
- **Use Case:** As a best practice, Hierarchical Forms are recommended when the user can fill out the information in any order, while Multi-step Forms are preferred for sequential data entry.

4. Data Model Integration and Efficiency

The effectiveness of the Constellation UI is tightly coupled with a well-designed Data Model. Key features like Primary Fields leverage the Data Model to automate and streamline UI configuration.

4.1. Designating Primary Fields

Primary Fields represent the most critical pieces of data for a specific Case Type or data object.

- **Function:** When a field is designated as Primary within the Data Model, it is **automatically displayed** on the Create, Edit, and Details Views associated with that Case Type or data object.
- **Default Behavior:** For Case Types, the Label and Description fields are designated as Primary by default.
- **Benefit:** This feature saves significant time and effort in UI configuration by ensuring that the most important information is always present in key contexts without manual intervention.

4.2. Inheritance via Reference Fields

The power of Primary Fields is amplified through data relationships. When a Case Type includes a reference field that points to a data object, an inheritance pattern is automatically established.

- **Mechanism:** The Primary Fields from the referenced data object are automatically passed into the Case Type's Views and displayed alongside the Case Type's own Primary Fields.
- **Streamlined Experience:** This creates a highly efficient workflow where Primary Fields are configured once at the data object level and are then automatically inherited by any Case Type that references it, requiring no additional configuration.

4.3. Data Model Best Practices

While it is possible to add new fields to a Case Type's Data Model directly from the Configure a View screen during UI development, this is not the recommended approach.

- **Pega Business Architect Best Practice:** The established best practice is to create the Data Model first using the **Data Explorer**. This approach encourages thoughtful design that considers field reuse and inheritance patterns across the application from the outset.

5. Module Context and Learning Objectives

The information synthesized in this document is derived from the Pega Academy module **"Displaying data with Constellation for Business Architects."** The stated learning objectives for this module are to enable a Pega Business Architect to:

- Describe the advantages of the Pega Constellation design system.
- Configure a View with different Field Types to capture and display various types of Case data.
- Create Views in Constellation that deliver a consistent user experience.
- Describe the role of Primary Fields in a Case.