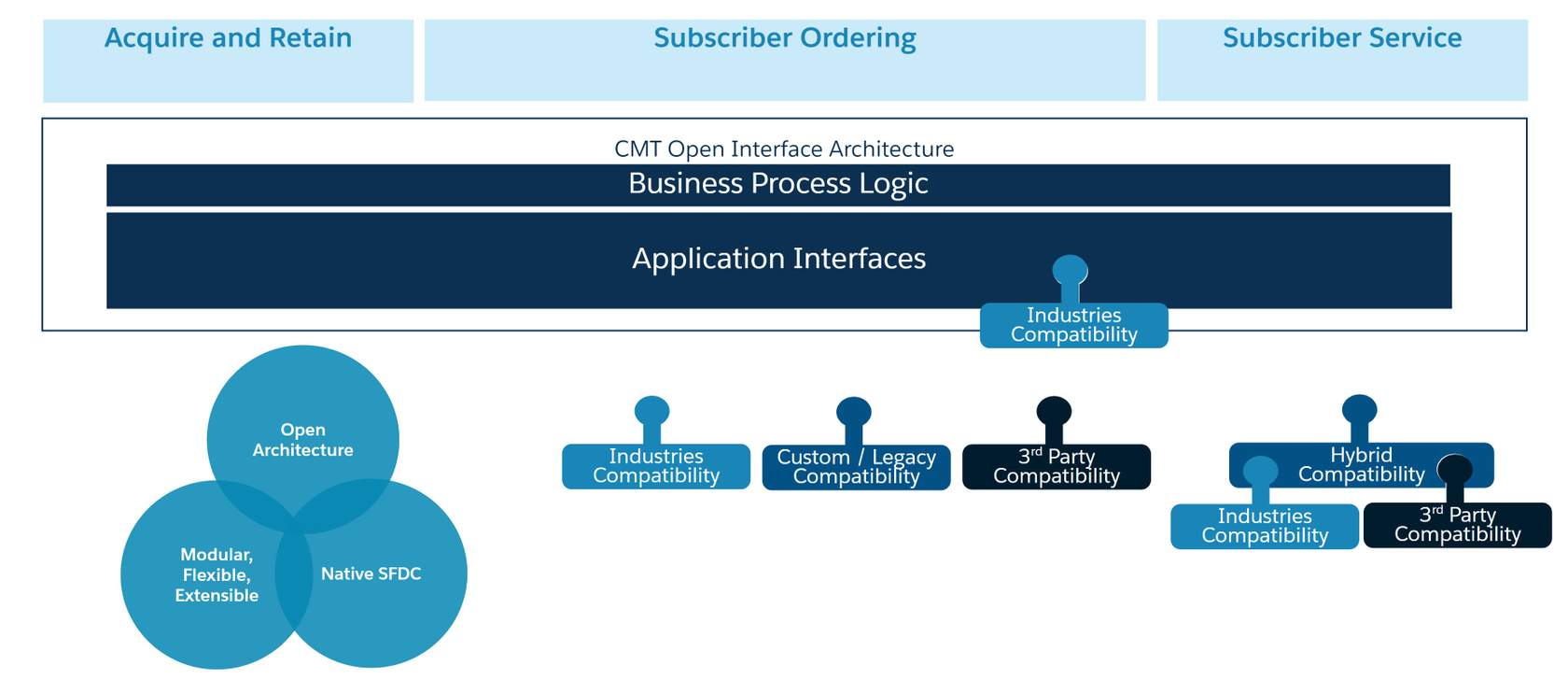
**The Implementation and Interface Paradigm**

The core of the Salesforce Industries open architecture is the interface and implementation paradigm. The Salesforce Industries Component Object Model (COM) delivers more than 60 interfaces and more than 100 implementations that provide an open, reusable, extensible, and modular approach to COM.

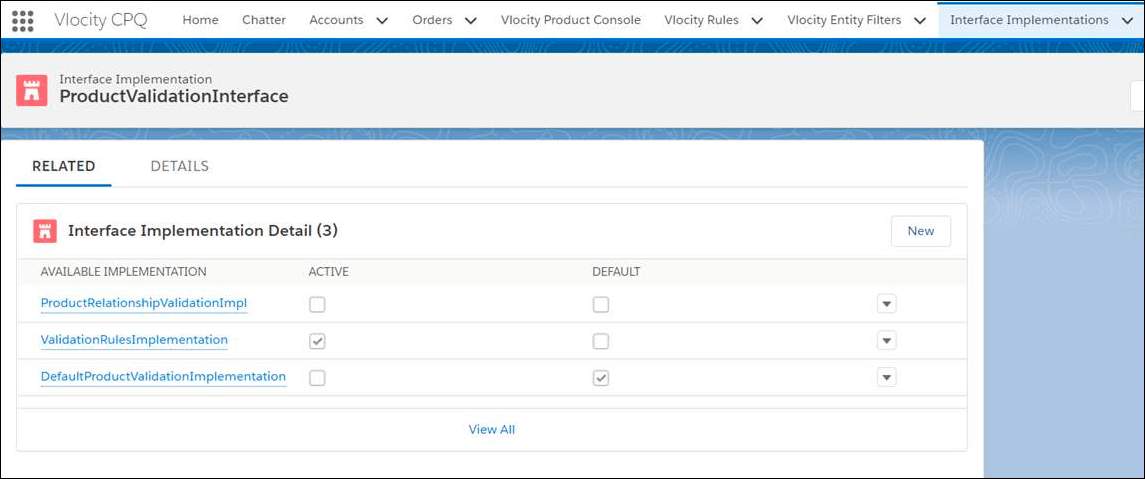
An **interface**is a call out to business logic. The **implementation** is that business logic. The interface asks the question and the implementation provides the answer. For example, the pricing interface communicates with the system to ask “how much do these items cost?” The implementation responds with the appropriate business logic to price the products.

**Choose the Right Tool for the Right Job**  
Salesforce Industries applications use interchangeable, reusable interfaces and implementations. You can customize the interfaces, replace them, or preserve your existing business logic. Salesforce offers multiple implementations for each interface. You can choose the implementation that best fits your needs or you can create your own implementation.

When used judiciously, you can define interfaces once and deploy them many times to consistently maintain and apply business logic.



**CPQ Implementations: Multiple Options for Each Interface**



*Interface Implementation Example: ProductValidationInterface*

Implementation and interface paradigms are central to the design of Industries CPQ. The process of creating an order is a complex series of actions. At certain points in this process, Industries CPQ makes API calls to the appropriate interfaces to implement business logic using Apex classes.

Your Salesforce Industries Administrator needs to switch on the correct implementation and interface that you want to be active. You can have only one implementation active per interface.

* Default means there is no logic, only a stub that allows the Cart to function without any rules.
* Custom means you write your own interface using Apex code.

**It’s Your Choice**  
Each interface and implementation has a trigger point.  
At the trigger point, you can call out to an external system and return information from that system. The implementation creates the output and sends it back to the interface.

1. **Default Implementations**
   1. Each interface has a “default” implementation that is essentially a stub or nonoperative “no-op” implementation. No-op interfaces and implementations receive input and return the same input as output without modification.
   2. Salesforce provides a variety of interface implementations that support rules. The default implementations allow the Cart to run, but they don’t invoke any rules. You must activate different implementations to apply rules.
2. **Salesforce Industries Implementations**
   1. Salesforce supplies a variety of OOTB implementations for each interface. You can choose the provided implementation that best fits your business needs.
3. **Custom Implementations**
   1. You can create your own implementations using Apex code.
4. **Hybrid Implementations**
   1. You can also create a hybrid solution that uses some Salesforce Industries implementations and some custom implementations. An interface can call external functions to resources outside of the Salesforce Industries application. These outside calls connect to other resources and pass parameters, and then return data to Salesforce.