**Product Modeling**

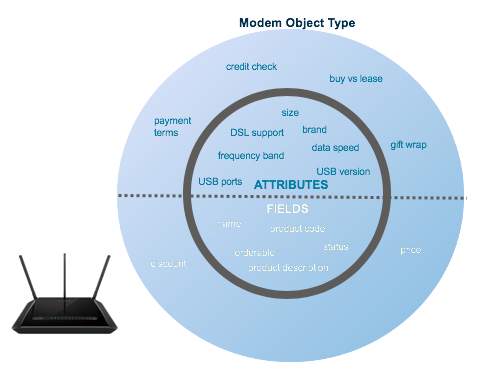
**Product Modeling Principles**

**Inside vs Outside**

When designing your product model, keep your product specifications pure as the driven snow.

Designate product details that fully describe the product as “inside” the product definition, and details that pertain to ordering, payment (buy vs. lease), shipping, etc. as “outside”. We're not trying to single anyone out here, but it's better to know who (what) should be inside the circle of trust vs outside. Am I right?

* Use product attributes to capture "inside" details that are specific to a product or a product family, and add custom fields for "inside" details that cross the entire product catalog.
* "Outside" details can be captured on the order, order line item or the offer specification.
* Organize attributes by functional attribute categories, such as Logistics or Physical Properties.
* Assign attributes and fields to object types, and allow product specifications to inherit them.



**Lego-Build it Bottom Up**

Oh, to be a kid again using your imagination to build a pirate ship or a castle out of Lego!  Building your product catalog uses the same concept, except you won't be screaming out in agony when you step on a product catalog Lego brick while barefoot. Ouch!

* Working from the lowest layer, build your product catalog in reusable "Lego" style blocks.
* Lowest level product entities should be tuned to external downstream systems, and highest level product entities should be tuned to the customer.
* Create multiple offers from a product specification or a combination of nested offers to drive the customer’s buying experience.



**Drive to the Edge**

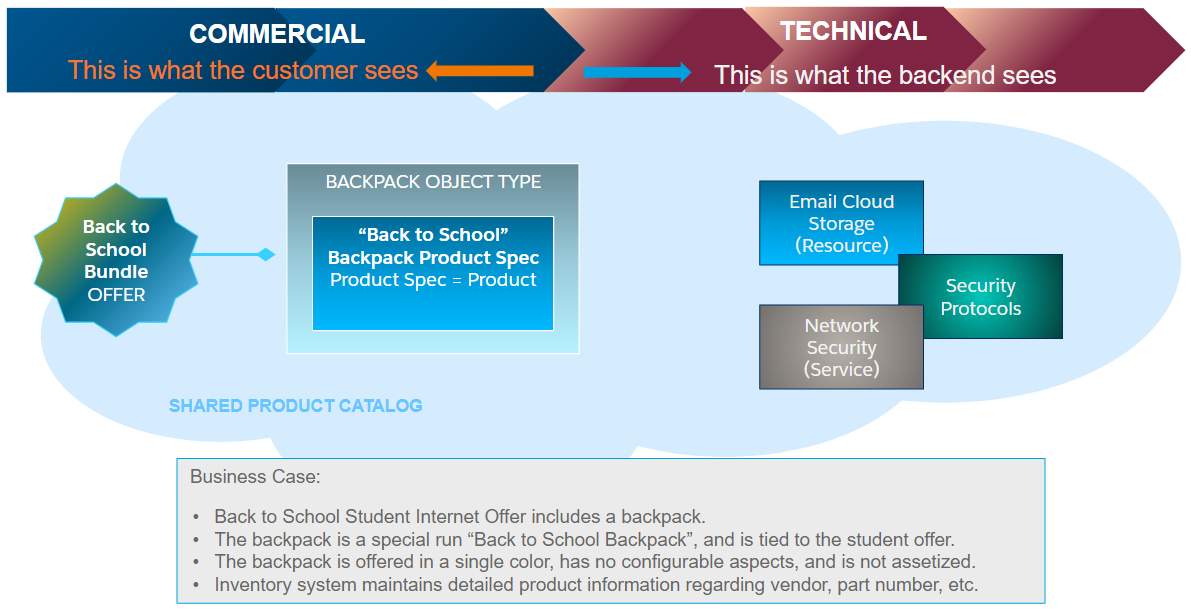
* Although Thelma & Louise was an entertaining movie to watch, we don't want you driving off a cliff! When designing your product catalog,*drive to the edge [of specificity]…without going over.*
* What do we mean by this? Products should be specified with “just enough” detail to allow customers to order them and send a fulfillment request to downstream systems, but without duplicating product data that is unnecessarily redundant.

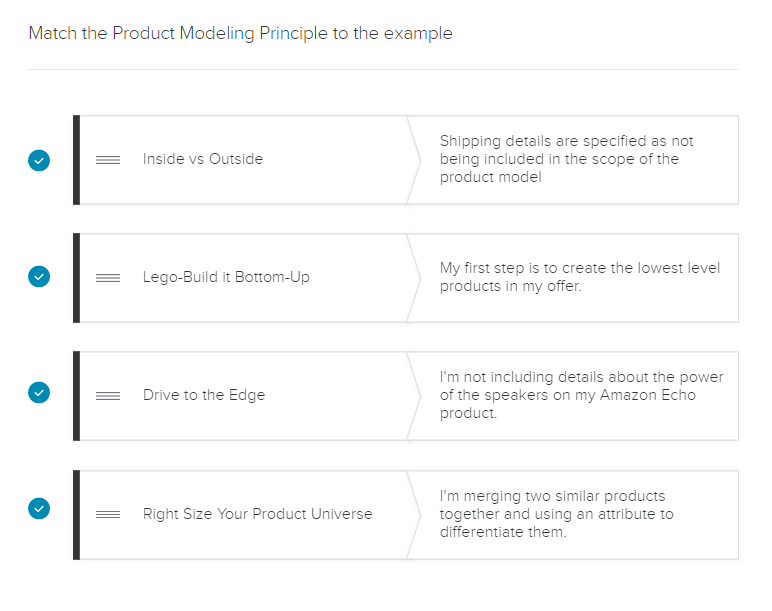


**Right-Size Your Product Universe**

The universe is vast with all its beauty. But your product universe should be sized to meet your current and expanding needs.

* Each product’s model should be individualized, and should only have the number of abstractions / levels that are truly necessary.
* To reduce product proliferation, consider amending or expanding existing product specifications rather than creating new.
* Complex products should be built with enough levels and mappings to insulate customers from backend logistics and unnecessary detail.
* Simple products should remain simple, despite the capability to create fully abstracted entities using the Shared Catalog (EPC).





**Fields and Product Attributes**

Product attributes provide a powerful and flexible way to extend products. They are stored in aggregate "blob" form in the JSONAttribute field on the Product object and are used extensively throughout Industries CPQ and Industries Order Management to:

* Filter products at run-time.
* Configure product specifications at run-time.
* Change product pricing using attribute-based pricing rules.
* In Industries Order Management, to map commercial products to technical products.

When should I use an attribute instead of a field?

Attribute

If the data element is:

* specific to a small set of records.
* to be run-time configurable (ie: configurable via the Cart).

Field

If the data element is widely used for products across the entire product catalog, create a new field on the Product Object.

**Remind Me - What's a Promotion?**

A promotion is a special type of product bundle designed to provide incentives to customers. Built on top of products, promotions may have one or more of the following features:

* Limited availability e.g. time frame or customer group (e.g. seasonal promotion).
* Limited products or product quantities.

If you're having a hard time remembering anything about Promotions and Discounts, please refer back to your *Build CPQ Solutions for Industries*course training materials for a refresher.

However, to get you started, we've included a handy checklist of questions you should ask when designing your promotion.

**Key Questions in Promotion and Discount Design and Implementation**

* How do you describe the promotion?
* How long is the subscription commitment?
* When does the subscription start/end?
* How does the subscription end?
* Which product/bundle(s) do you want to use?
* Cardinality: What is the minimum, default, and maximum quantity of the entire bundle?
* Cardinality: What is the minimum, default, and maximum quantity of each child product?
* Which individual products' prices do you want to adjust?
* Do you want to use an adjustment or override the price?
* How long do you want the lower price of the product to be in effect?

**Deciding which rule type to use**

Need some help with rules? The tables below should help with your decision-making process, but if you need more help, refer back to your training materials for the Context Rules and Advanced Rules *Build CPQ Solutions for Industries*courses.

| **When should I use each rule type?** | | |
| --- | --- | --- |
| **Stage** | **Context Rules** | **Advanced Rules** |
| Orders | ✔ | ✔ |
| Quotes | ✔ | ✔ |
| Opportunities | ✔ | ✔ |
| Order Line Items | ❌ | ✔ |

| **Which objects use Context Rules/Advanced Rules?** | | |
| --- | --- | --- |
| **sObjects** | **Context Rules** | **Advanced Rules** |
| Products (Offers) | ✔ | ✔ |
| Promotions | ✔ | ❌ |
| Contracts | ✔ | ✔ |
| Price Lists/Price List Entries | ✔ | ❌ |
| Pricebook/Pricebook Entries | ❌ | ✔ |
| AdjustmentData Virtual Object | ✔ | ❌ |

|  |  |  |
| --- | --- | --- |
| **Why do you need this rule?** | **Context Rules** | **Advanced Rules** |
| To check product compatibility | ❌ | ✔ |
| To check product availability | ✔ | ✔ |
| To check customer eligibility | ✔ | ✔ |
| To auto-add products based on other products in the Cart | ❌ | ✔ |
| To auto-add products based on the applied promotion | ✔ | ❌ |
| To apply a penalty for a promotion or contract cancellation | ✔ | ❌ |
| To select a single price when a customer is eligible for multiple prices (tightest match) | ✔ | ❌ |
| To restrict manual pricing adjustments | ✔ | ❌ |
| To use with Digital Commerce cacheable APIs | ✔ | ❌ |