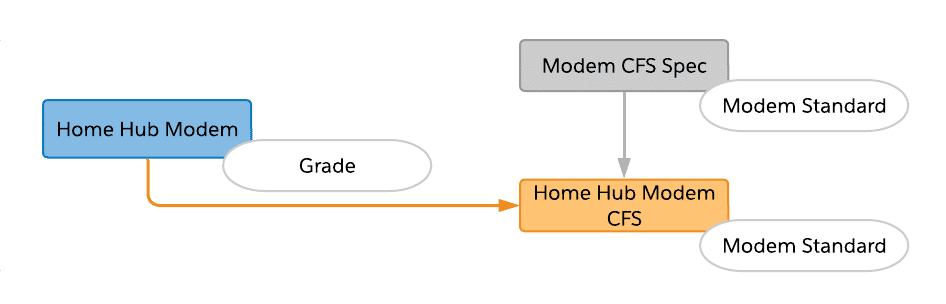
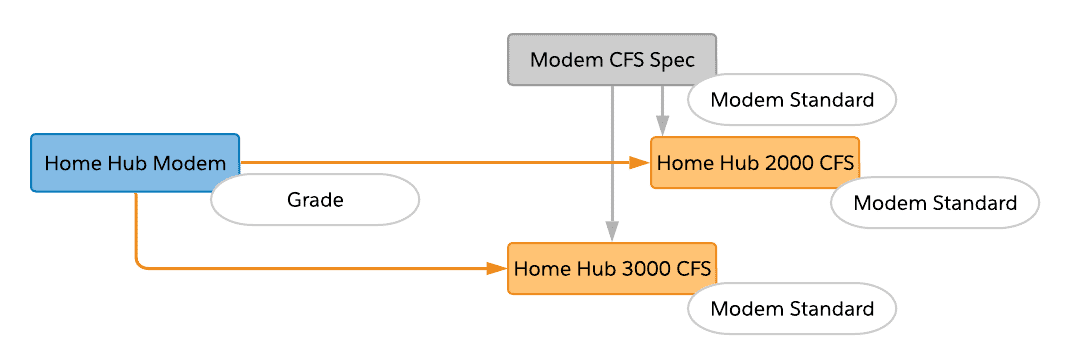
**Types of Decomposition Relationships**

**One to One (1:1)**



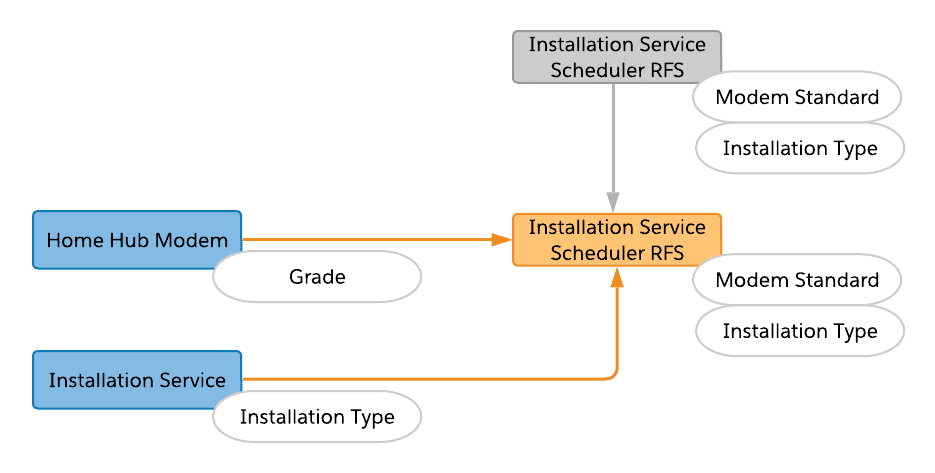
One-to-one relationships are the most common and straightforward type of decomposition relationship. In this example, the Home Hub Modem (commercial product) decomposes to the Home Hub Modem (technical product).

**One to Many (1:M)**



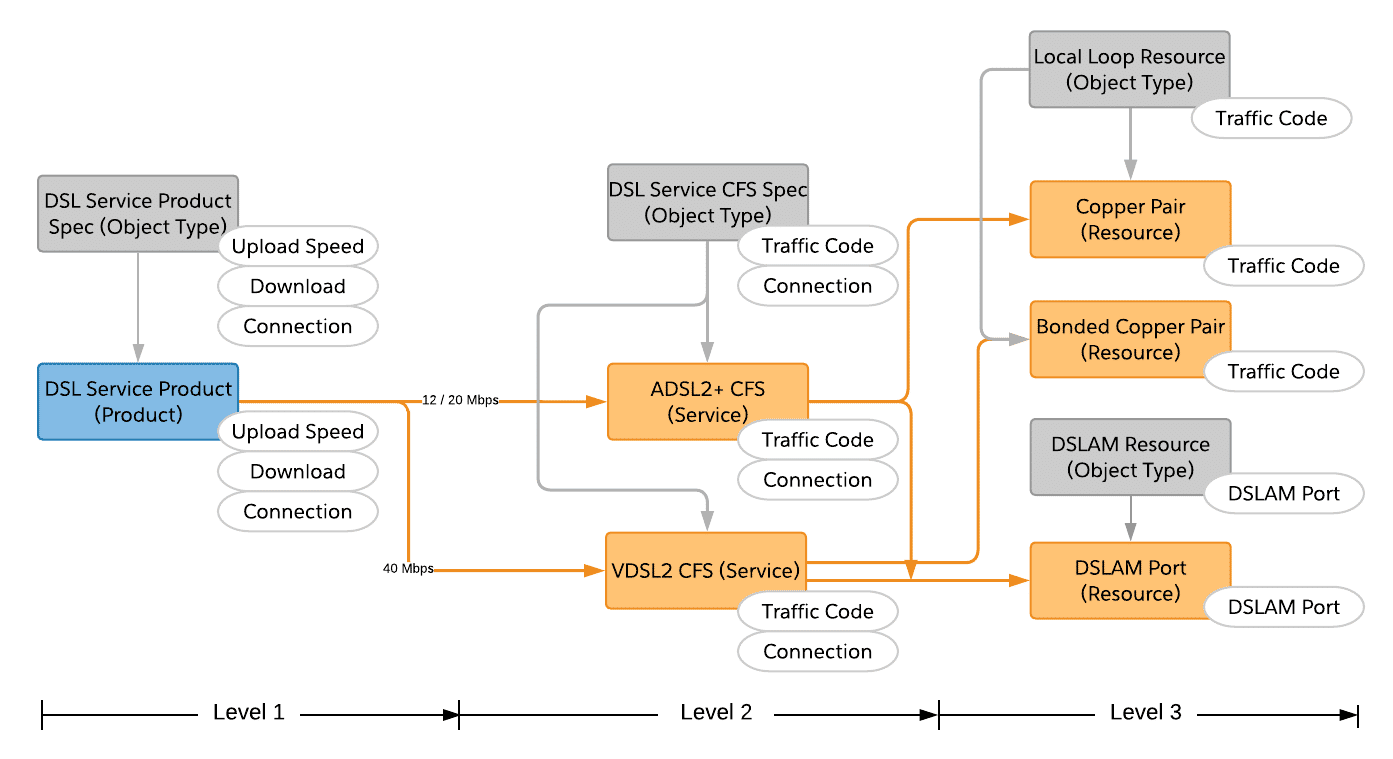
Sometimes a single commercial product should decompose to one of several different technical products. This simply means that one source product will have multiple decomposition relationships. In this example, based on what the Grade attribute is set to (Good or Best), the relationship is set to Model 2000 or Model 3000.

**Many to One (M:1)**



Many to one relationships use multiple decomposition relationships in combination with the Scope function. In M:1 relationships decomposed fulfillment requests can be consolidated, which is more efficient when communicating with fulfillment systems.

**Multi-level Decomposition**



Lastly, you can also create multi-level decomposition relationships. They are comprised of 1:1, 1:M, and/or M:1 relationships. As a best practice, a maximum of four levels is recommended. Remember, sometimes simpler is better. Too many levels bring complexity and complexity can result in confusion.

Again, decomposition relationships map fields and attributes of commercial products to the attributes of technical products. Notice that the source product data can be a field or an attribute, but the destination product data must be stored as an attribute. That is, a technical product contains only attributes (not fields).

There are two more things you should know about before building out decomposition relationships:

1. Conditions
2. Mapping Rules

**Conditions and Mapping Rules**

**Conditions**

A condition rule places a condition on the decomposition relationship. Industries Order Management initiates decomposition of an order item only if the specified condition evaluates to true. Otherwise, Industries Order Management skips the decomposition relationship.

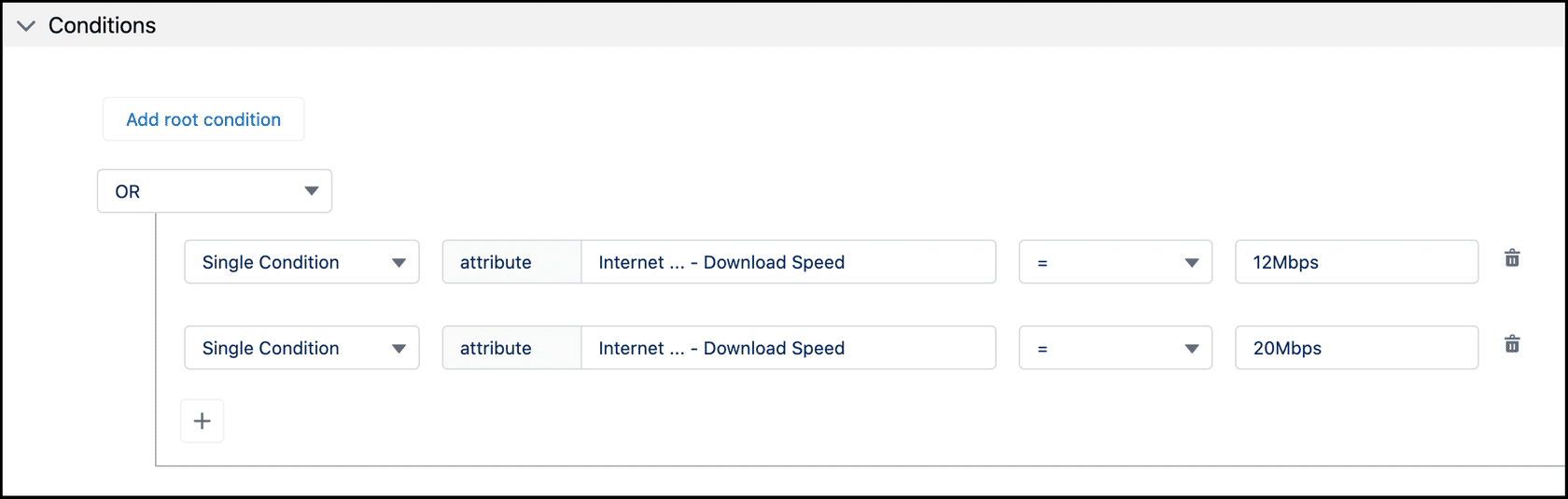
**Similar to an SQL WHERE Clause**

Condition rules function very similar to an SQL Where clause. However, rather than restricting the data set returned from a query, they are used as control logic for the decomposition process. Only decompose if <whatever> is true.

**Example Use Cases**  
A customer selecting a home internet package along with installation options serves as an example use case.

* Map upload and download speed limits depending on the package they select: Gold, Silver, or Bronze
* Determine if a home installation is required or not (based on whether they purchased their own router, or went with a lease option)

How to specify Conditions in the UI is pretty straightforward.



**Mapping Rules**

Mapping rules define the information passed from the source product to the destination product, and thereby to downstream systems. Although you may want to pass the attribute or field data "as is", you don't have to.

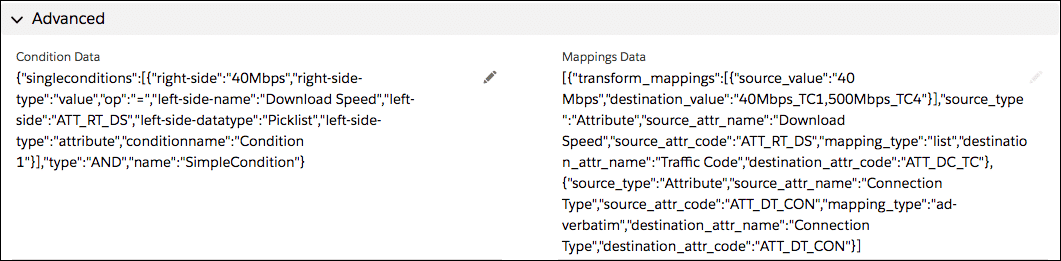
There are three different types of Mapping Rules.

1. **Ad-verbatim** - Copies the value from source to destination, without changing it. This simply passes data through untouched.
2. **List** - Transforms the source value to a new destination value. For example, map a "Gold" or "Silver" plan to a specific throughput speed such as "100Mbps" or "80Mbps".
3. **Static** - Set a predefined value for the destination attribute.

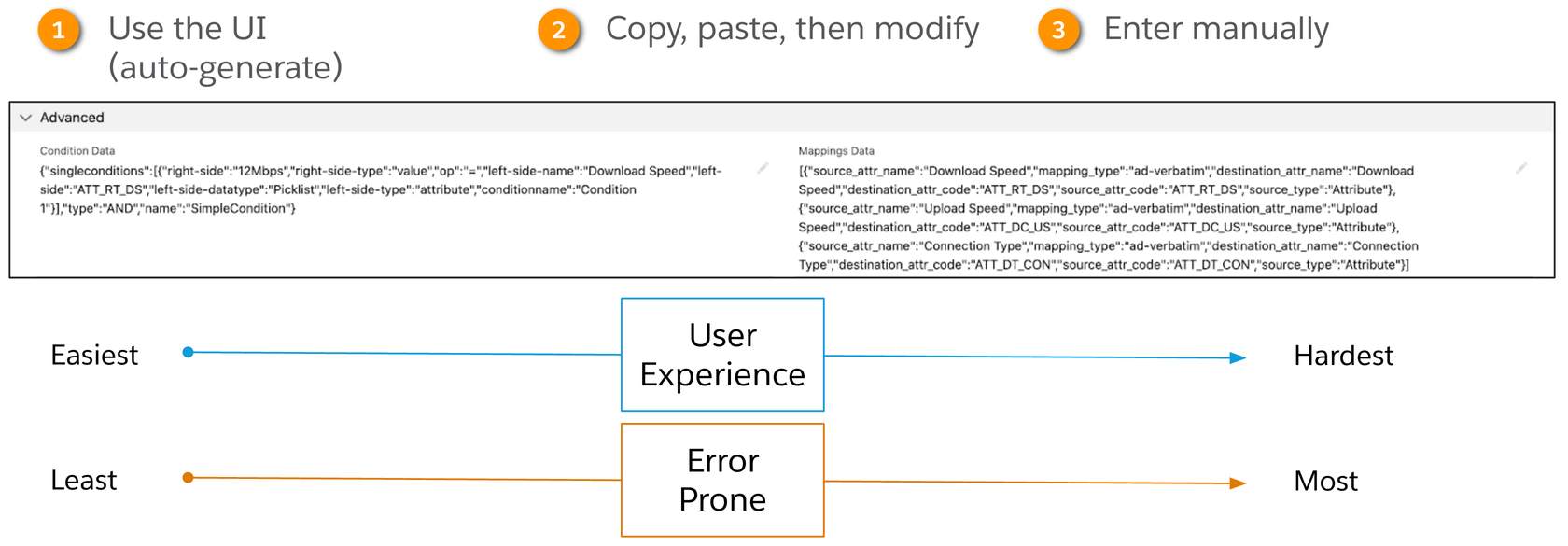
# Using Order Decomposition

**Data Format**

Both Condition and Mapping Rules are represented as JSON data in Industries Order Management. After configuring the Condition and Mapping Rules for your decomposition relationship, the **Advanced** section reveals the JSON blob.



Notice the pencil icon for editing the JSON. Although you can edit the JSON directly, it is not recommended for relative beginners with respect to JSON or order management. It's simply too error-prone. Thankfully, Industries Order Management generates the JSON for you when you define Conditions and Mapping Rules.



Summary of **three** ways to populate Condition and Mapping JSON data

**Invoking and Viewing Decomposition Relationships**

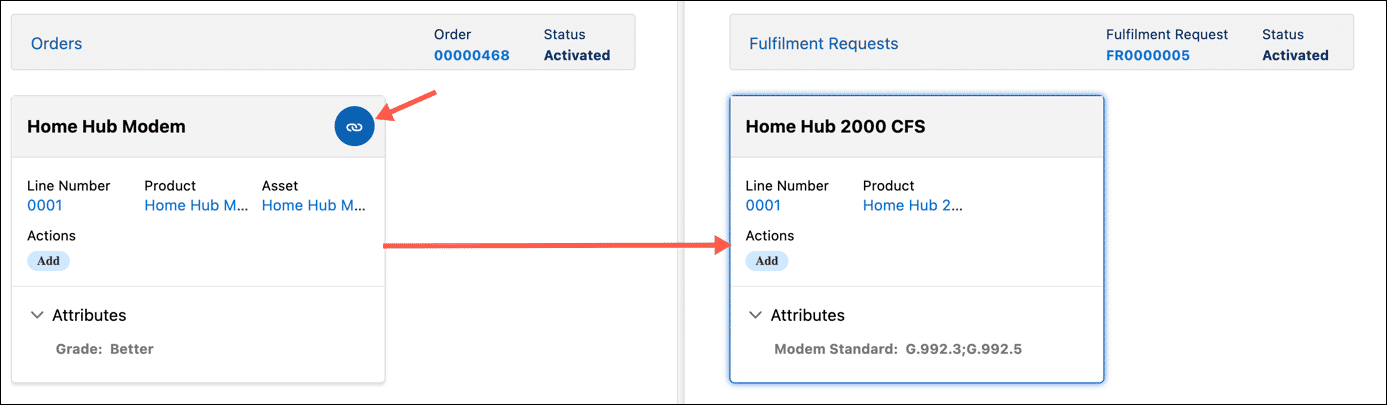
Having learned the basic concepts of the order decomposition process, you may be wondering what exactly this looks like, and how do you invoke the process from within Industries Order Management.

**Invoking**

* **Training Environment**(training org) - Click the **Decompose Order** button in the Cart. A manual step is helpful for learning, testing, and debugging purposes.
* **Production Environment** - In production, the decomposition process is invoked automatically when an order is submitted (and automatically triggers the orchestration process as well).

**Viewing**  
**Training Playground**- The**Decompose Order**button shows you the decomposition relationship when it completes. However, there is also a **View Decomposition** option provided. Orders can only be decomposed one time. **View Decomposition** allows you to see previously decomposed orders.

* **Training Playground**- The**Decompose Order**button shows you the decomposition relationship when it completes. However, there is also a **View Decomposition** option provided. Orders can only be decomposed one time. **View Decomposition** allows you to see previously decomposed orders



**Scope**

Setting the **Scope** field on the product entity allows the decomposition results to be reused, instead of re-instantiated. The two most common scope settings are:

1. **Order Item** (default)
2. **Account**

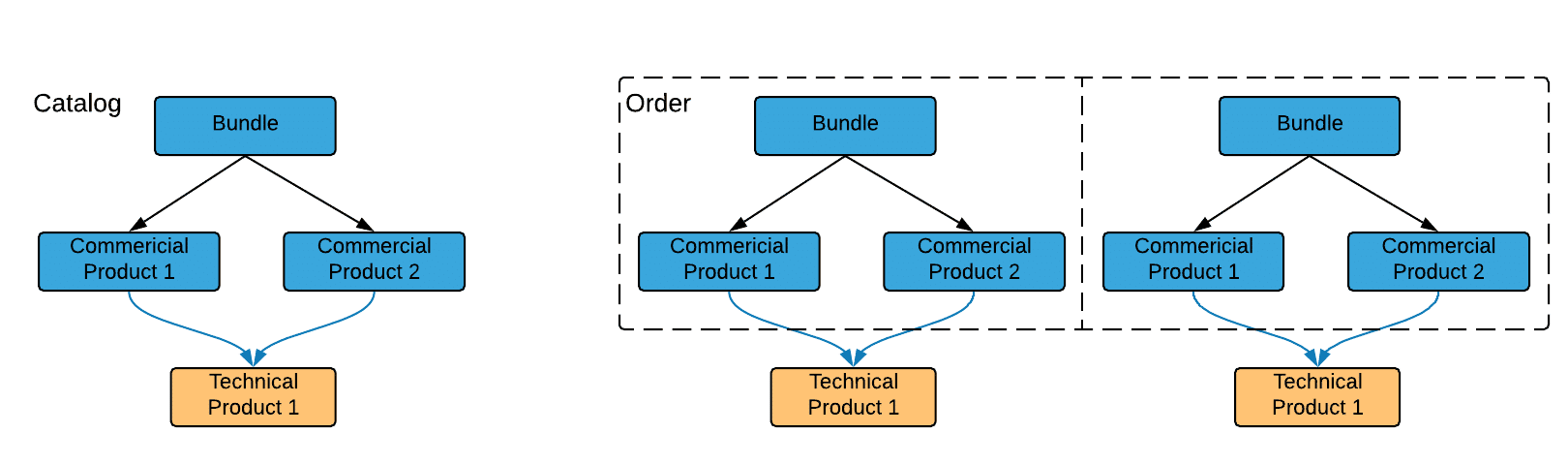
Industries Order Management generates a number of Fulfillment Request Lines (FRLs) for downstream order items as a result of the decomposition process. Setting the scope to **Order Item** or **Account** changes the number of fulfillment requests in the decomposed order.

|  |  |
| --- | --- |
| 📝 | The default setting for the Scope is Order Item. Note that "--None--" is equivalent to Order Item as well. |

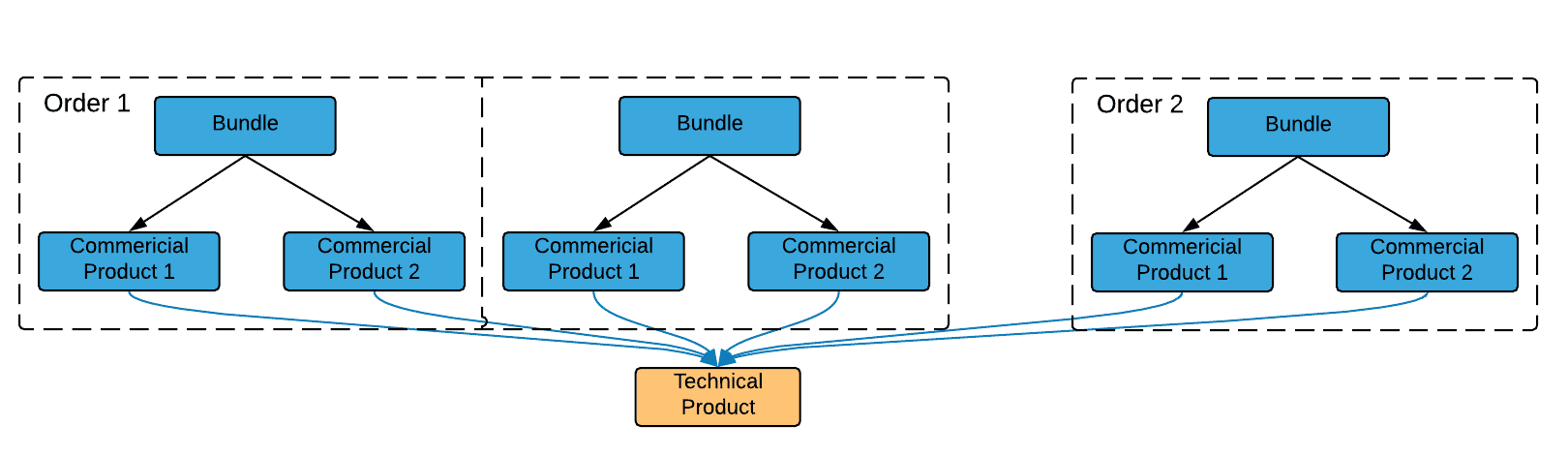
**Order Item Scope**

The diagrams below help illustrate the different behaviors based on the product scope setting.

Based on the product catalog (below left), and the scope set to Order Item, an order for a mobile product bundle results in two fulfillment requests when decomposed. Click on the hot spots below to see additional information



**Account Scope**  
If the scope is set to Account on the product and decomposed, it results in a single fulfillment request. This of course streamlines the processing and communications with downstream systems.



The example above is better suited when a single piece of equipment can support multiple services for the same customer (regardless of the number of orders). Order 1 includes two bundles with TV service, Broadband and a VOIP line. Order 2 is added later (for the same customer) for an additional VOIP line. However, the existing modem can support all services, so there is no need to ship another modem. Decomposition does not produce another technical product (like it would have if the product Scope was Order Item).

**Product Class**

By default, when you create a new product using the Product Console, the Record Type is set to Product. Technically speaking, Product Class is an abbreviated way to refer to a **product with the Record Type set to Class**.

Product Class is not simply a commercial product or technical product, it's a mechanism used by Industries Order Management to categorize certain products in order to simplify the order decomposition configuration process.

**Use Case**

Just knowing what a Product Class is doesn't help with the use case. What is it used for? What challenges does it help solve? You can use Product Class to identify products that have similar behavior in decomposition.



**One of these things is not like the other**

But that does not mean you can't identify similar puppies (or products in our case), and use that to simplify the task at hand. (Which is either dog grooming, or order decomposition, I forget which.)

For example, similar phones might share the same fulfillment process. Or a quad-play home internet package might include similar routers, that require scheduling a technician to install it on-site for the customer.

At this point, you have had a chance to create several decomposition relationships, which is relatively easy when you don't have that many products. Now imagine that your product catalog contains thousands or tens of thousands of products. Setting up a decomposition relationship for each and every product is daunting, hard to maintain, and can be error-prone. Fortunately, Industries Order Management provides a better way - using Product Class!

**Using Product Class**

Implementing decomposition by Product Class includes three main steps:

1. Create the Product Class
2. Associate similar commercial products with the Product Class
3. Create a single decomposition relationship