**Integration Discovery Questions**

For each integration:

1. What is the use-case?
   1. What is this integration’s expected functionality?
   2. What does the customer expect this integration to do?
   3. What data set is in scope?
2. What is the external system?
   1. What version?
   2. Is it cloud-based or hosted on the customer’s network?
   3. Does it have an accessible API?
      1. If so, what kind? (e.g., REST)
      2. Will Lockpath have access to the API resources/documentation required to build this integration?
      3. What payload formats can it accept/deliver? (e.g., XML, JSON)
   4. Can it export desired data in a flat format?
      1. If so, in what formats? (e.g., CSV, XML)
3. Where is the data going?
   1. From an external system to Keylight?
      1. Which tables, fields, etc.? Be explicit.
   2. From Keylight to an external system?
      1. Which tables, fields, etc.? Be explicit.
   3. Bi-directional sync?
      1. What attributes are in scope?
         1. Have the applicable objects and attributes been mapped between the two systems?
            1. Do the mapped attributes share the same properties? (e.g., does a numeric field in the source system map to a numeric field in the target system?)
      2. In the event of synchronization conflicts, which system is the system of record?
4. Is the integration only supposed to create new records, or is it expected to create new as well as update existing records?
   1. What fields/attributes need to be continually updated?
5. How frequently does the source data update?
6. When does the source data need to be transmitted to the target system?
   1. How frequently?
   2. Under what conditions/criteria? (e.g., when workflow conditions are met? when a record-level value condition is met?)
7. Does the customer have development/test resources available for the external system that Lockpath can use to build and test the integration?
   1. Resources include, among others:
      1. SMEs
      2. Technical documentation
      3. Development/test environments