IdP Proxy (proxying either IdPs or SPs) as a Metadata Entry

We have a few use cases where we will need to deploy an IdP Proxy. The use cases mostly revolve around vendors that can only communicate with a single IdP per site, but where we have multiple IdPs that will authenticate users. The IdP Proxy is in place to make these SPs see a single IdP instead of many. We have some other use cases that have slightly different justifications for the existence use of the IdP Proxy, but the underlying issues from a "New Entities" standpoint are similar for all the use cases.

Conceptually, the use cases break down into two categories of issues:

1. **IdP Proxy "in front of" an SP.**
    a. In this case we want to register the IdP Proxy as an SP entry, even though there is(are) a distinct SP(s) being served by the IdP Proxy. In most cases, The SPs being protected will be unregistered (i.e., no metadata in InCommon)
    b. Are there issues registering such an "intermediary" SP in InCommon?
        i. I presume that in general it's okay to register the IdP Proxy as an SP, as the actual architecture of whether the SP is installed locally (on the vendor application servers) or remotely (at a "blessed" IdP Proxy) doesn't affect the overall business integration.
    c. What would be the operating expectations of an SP that stands in front of *multiple* external vendors?
        i. There is a risk that the IdP Proxy could use its access to data from IdPs to begin releasing data to additional SPs. (Technically any SP could do this, but by nature of being an IdP Proxy, it may make the Proxy operator less aware of the need to make the SPs visible to external IdPs)
        ii. Do the existing POP and other requirements of registration address these concerns?
2. **IdP proxy "in front of" multiple IdPs** (that may be registered in InCommon themselves)
    a. Vendor SPs may want to connect to our IdP Proxy using InCommon metadata for their configuration.
    b. Is it possible for such an IdP Proxy to be registered in InCommon? What additional requirements would an IdP Proxy be held to?
    c. Again, because this IdP Proxy has the ability to issue assertions that appear to come from a single SP (to the IdPs), there is the risk of bypassing the proxied IdPs' release practices with an IdP Proxy in the mix.

In my use case discussions, regardless of whether the IdP Proxy gets listed in InCommon, my expectation is that we would deploy such proxies with targeted entityIDs. That is, to avoid the potential issues called out in 1.c. and 2.c., if we have two applications that need to be proxied (and we do!), we would configure the IdP Proxy in such a way that the SP entityIDs seen by the proxied IdPs are distinct and still make visible to the IdPs what specific SP /service is being accessed.