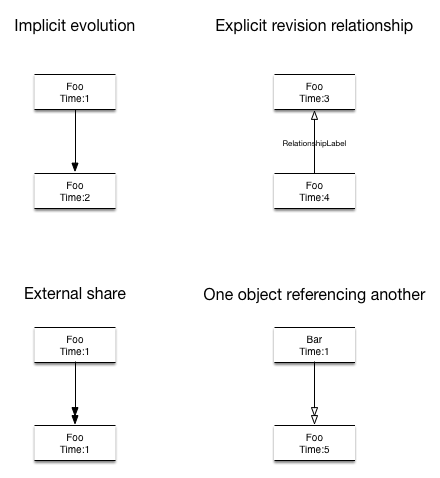
This document simply gives some visual representations of some of the scenarios/issues relevant to Revision and Revocation in STIX, particularly focused on the Option #6 approach.

Sean created these to help guide his thinking. ☺

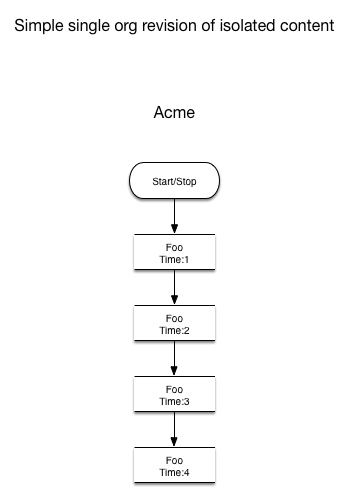
In the following diagrams:

* Don’t take the Time numeric values as literal. They are just for positioning within the diagrams.
* The local/internal revisions assume an implicit model (just timestamp updates without explicit revision relationships)
* The label values on explicit revision relationships is not intended to be normative. They are just some possible example values based on likely scenarios.
* The use of a arrows follows these conventions

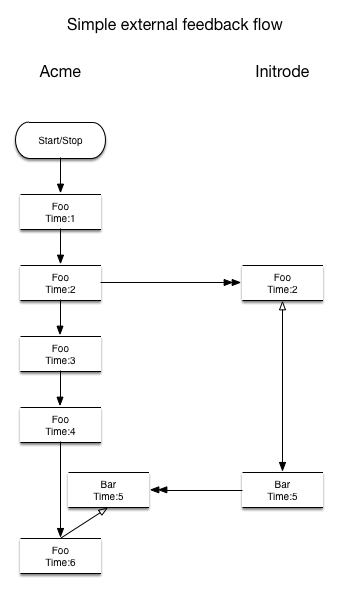


# Evolution Scenarios

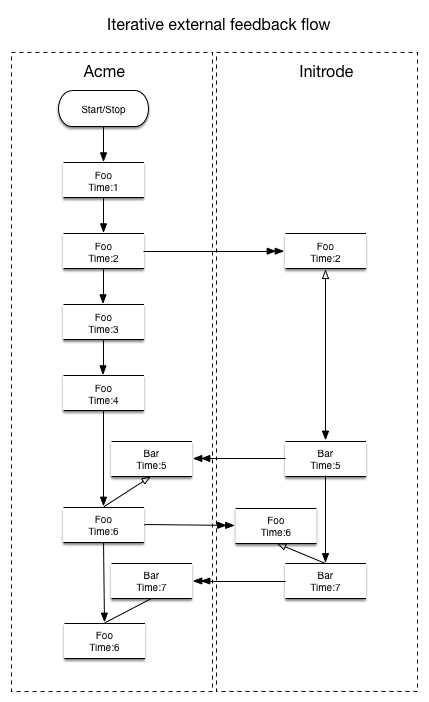
## The most basic content evolution scenario where one organization updates its own content internally.



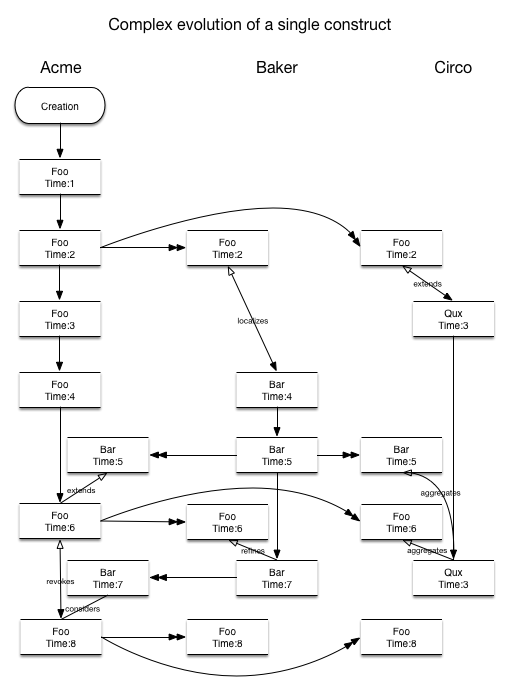
## Simple evolution of content with external sharing and feedback



## Iterative evolution of content with external sharing and iterative feedback



## Complex Multi-Party Evolution of content with sharing



# Referencing Scenarios

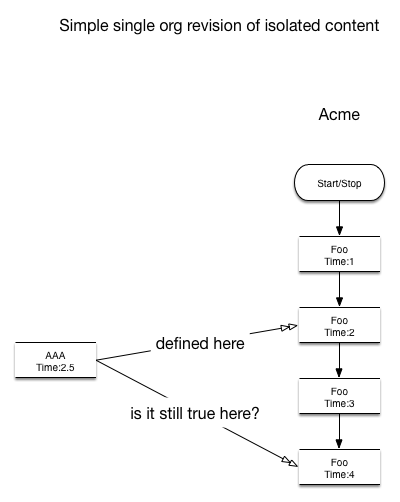
## Some Fundamental Questions with References

When references are made to content that is evolving, how should those references be interpreted and/or maintained? For some cases, it may be useful to treat such a reference as a reference to the evolutionary chain/cluster of the object being referenced such that the reference still holds true as the content evolves (presumption is that the meaning of the referenced object is not really changing).

It is also certainly the case that in some situations the evolution of content does change its fundamental meaning, especially in the context of some defined relationship. In these cases, it would be necessary to support the reference to a specific version of an object and not to its further evolutionary versions.

One of the perspectives/models may be more applicable for some types of content than the other.

It seems necessary to support both models? Your thoughts?



### Perspective 1: General reference to concept and its entire evolution

This can be supported in Option #6 through use of only @idref

### Perspective 2: Reference to a specific version of an object and not to its future evolutions

This can be supported in Option #6 through use of both @idref and @timestamp\_ref

NOTE: The following diagrams do not show the object making the reference but rather to the constructs that are intended to be referenced. Green indicates possible, red not.

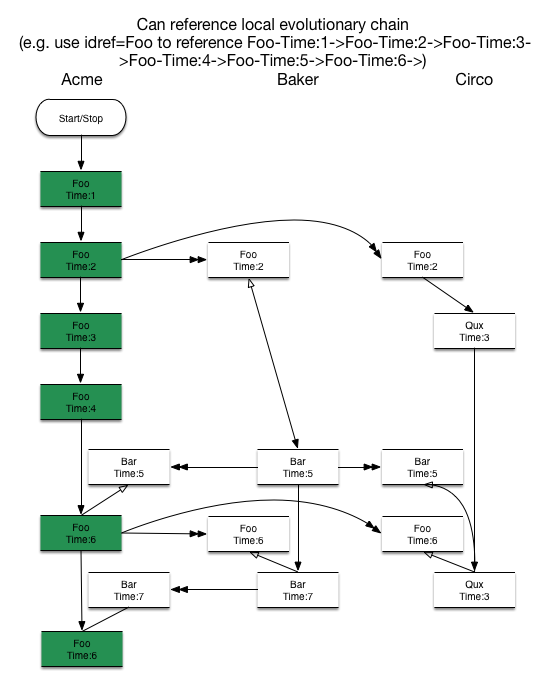
## Reference to a specific version of an object

This is supported in Option #6

## Reference to a local evolutionary chain

This is supported in Option #6

Open Question: If you reference a local evolutionary chain this way how should it be interpreted (the full chain?, the latest version in the chain?, or the orginal version in the chain?)? I have heard different parties in this discussion take each of these positions. Do we need an explicit way to specify which is intended?



## Reference to a global evolutionary chain

This is not supported in Option #6. Does anyone disagree?

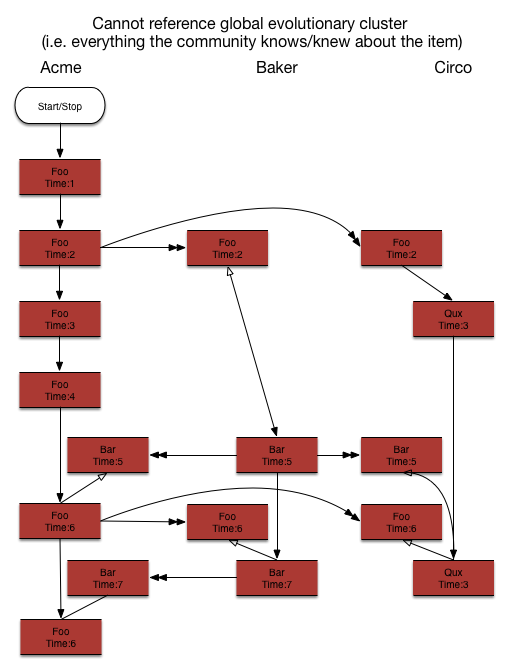
This seems useful. Is this needed?



## Reference to global evolutionary cluster

This is not supported in Option #6. Does anyone disagree?

This seems useful. Is this needed?



# One potential issue without History construct or explicit local/internal revision relationships

There is an open question on whether or not to include the History construct for SITX 1.1. I am not asserting an absolute opinion either way but I do believe there are potential issues if it is not included. One simple example is shown in the diagram below. If there is no History construct and implicit internal evolution (just timestamps and no explicit revision relationships) is used then Acme would have no way to determine that Bar-Time:5 derived from Foo-Time:2 that they initially sent out. They would have no visibility into the revision relationship between Bar-Time:4 and Bar-Time:5. This could be resolved either through use of the History construct or through explicit revision relationships for local/internal evolution (which we cannot require if we want to support Bernd’s simple common case).

