



# MESHERY

## Connections State Machine

*Design Specification*

<b>Design Prologue</b>	<b>3</b>
<b>Design Goals</b>	<b>3</b>
<b>Design Objectives</b>	<b>3</b>
<b>Functional Architecture</b>	<b>4</b>
State Machine Diagrams	4
States and their transitions	6
FAQ	6

# Design Prologue

Any resource that Meshery is aware of can be considered as a connection. For example K8s clusters, applications within K8s clusters managed by Meshery, Meshery adapters etc. Currently, there isn't a way to accurately track and manage the state of these connections. Hence we require a connection state machine within Meshery that makes it convenient to observe, manage and track the state of these connections.

MeshSync is a custom Kubernetes controller that provides tiered discovery and continual synchronization with Meshery Server as to the state of the Kubernetes cluster. The connection state machine would assist in managing the state of cluster objects post discovery by MeshSync.

## Design Goals

The designs in this specification should result in enabling

1. **Managing the state of K8s clusters**
2. **Managing the state of components discovered by Meshsync within a cluster**
3. **Extensibility to configure workflows to be triggered based on transitions in the state machine**

## Design Objectives

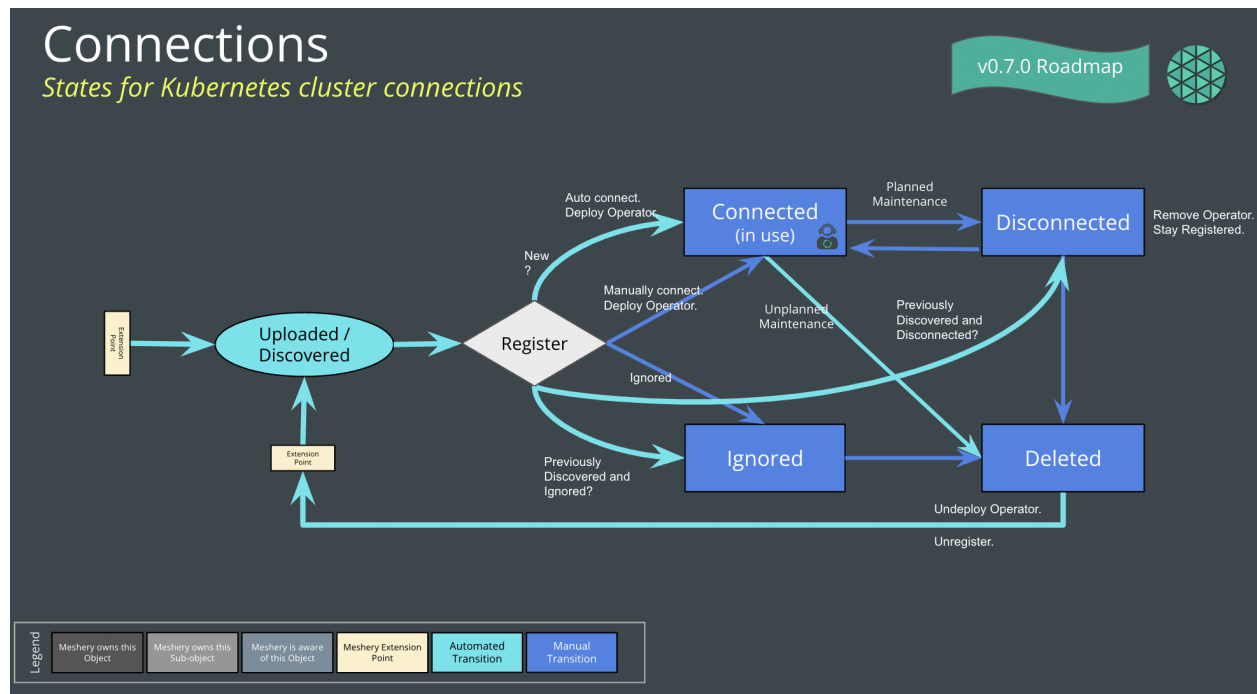
The designs in this specification should result in these specific functions:

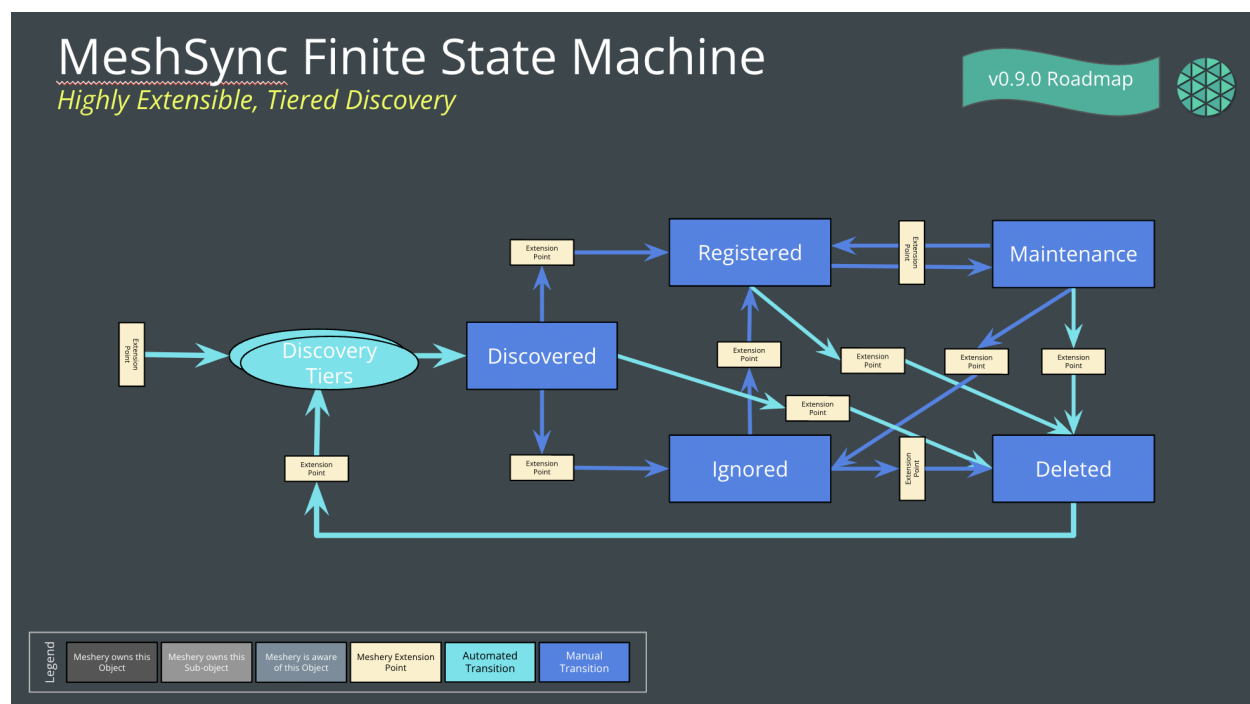
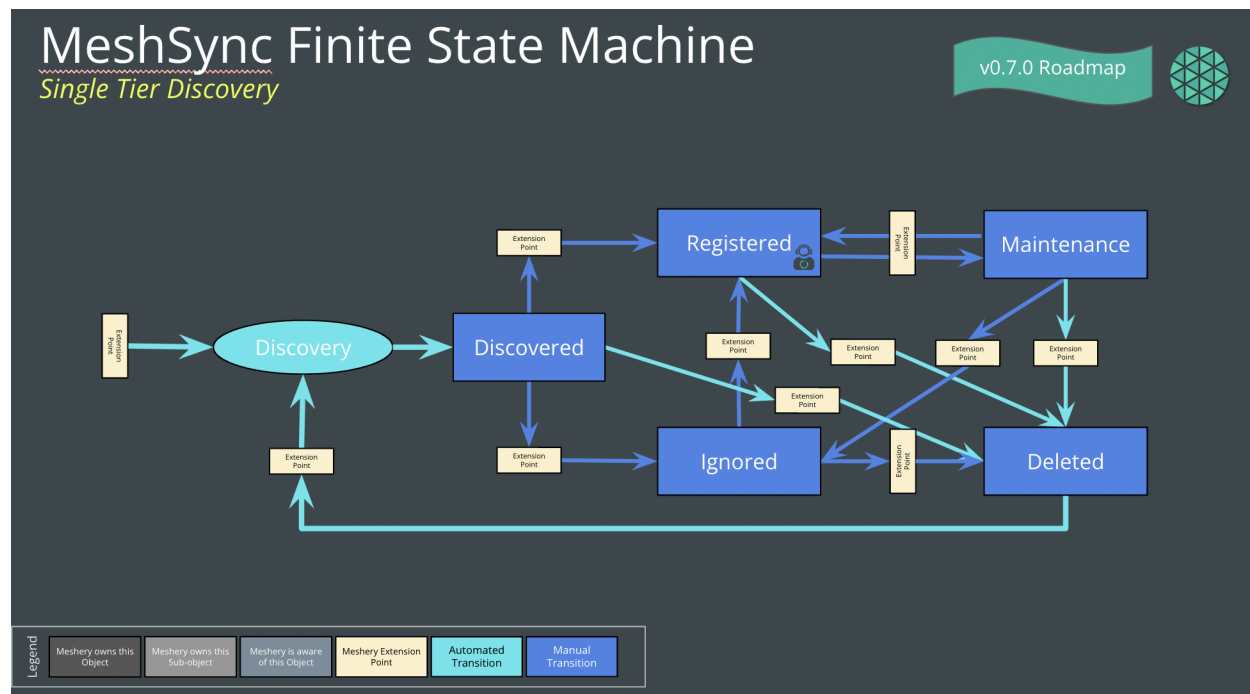
1. Management of Meshery Operator deployment in K8s clusters - Currently Meshery Operator is installed in all the connected clusters automatically once Meshery server detects those clusters. There needs to be a more granular control in deploying and removing operators in clusters detected by Meshery making it convenient for a User to choose what clusters they would like to register for use and which ones they would want to ignore.
2. Ability to register/ignore resources discovered by Meshsync - Clusters may contain a large number of resources, all of which may not be relevant to be shown in the UI when working with Meshery. The state machine would make it convenient for a User to register and ignore select resources.

3. Credential management for mesh models - Connections may require credentials for authentication to transition between the discovered and usable state.
4. Ability to extend the state machine with workflow engines - Allowing the extension of the state machine with a workflow engine like [Temporal](#) would enable users to configure custom workflows to be triggered as a response to state events.

## Functional Architecture

### State Machine Diagrams





Tiered discovery involves discovering cluster components in stages to make discovery efficient in cases where the number of components in a cluster are very large.

## States and their transitions

State	Description	Supported Object Type	Transitions To	Transitions From	Meshery UI Location	Meshery UI Component	Meshery UI Action
<b>Discovered</b>	Discovered, but not administratively processed		Registered Ignored Not Found	Not Found (only by being re-discovered)	MeshSync Browser (not implemented, yet)	Active row in data table	Register, Ignore
<b>Registered</b>	Administratively processed, online for use		Maintenance Not Found	Discovered Ignored Maintenance	Settings → Connections	Active chip w/icon Object name and version in tooltip	1. On-click: ad hoc connectivity test 2. On-click of "x": Disconnect
<i>Connected</i>	<i>Same as Registered for now</i>		<i>Same as Registered for now</i>	<i>Same as Registered for now</i>	<i>Same as Registered for now</i>	<i>Same as Registered for now</i>	<i>Same as Registered for now</i>
<b>Ignored</b>	Administratively processed, explicitly excluded from use by solution		Registered Not Found	Discovered Registered Maintenance	MeshSync Browser (not implemented, yet)	Inactive row in data table (gray background)	Discovered Registered Maintenance
<i>Maintenance</i>	<i>Administratively offline/unavailable</i>		<i>Registered Ignored Not Found</i>	<i>Registered</i>	<i>Not planned</i>	<i>Not planned</i>	<i>Not planned</i>
<b>Disconnected</b>	Previously known, but missing from the most recent discovery		Discovered (only by being re-discovered)	Discovered Registered Ignored Maintenance			
<b>Deleted</b>	Intentionally removed from management						
<b>Not Found</b>	Manually entered connection, but never connected		Discovered		Settings → Connections	In active chip w/icon. No tooltip.	

## FAQ

- **Difference between 'delete' versus 'not found'**
  - A user intentionally deleted a connection, explicitly removing it from Meshery's management / field of view. This is distinctly different from a kubeconfig with 5 contexts inside, 1 of which has been connected and the other 4 of which are not found yet.
- **Difference between 'ignored' versus 'disconnected'**
  - A user intentionally chooses to have Meshery ignore a given context, explicitly leaving in Meshery's field of view, but identifying it as a context not to manage. This is distinctly different from a cluster that Meshery was managing, but has been turned off and now Meshery is disconnected from the context.
- **Does the connections die when Meshery Server tear down?**
  - Except Meshery Connection all connectitons are left intact.