

## Open Container Initiative Scope Table

This table will be a live document on the OCI website once the charter has been finalized.

### NOTES FOR TABLE

The use of the term “layer” in this section refers to the layers of the software stack that would be developed to support a container runtime and any additional higher level components that might be added on top of the core OCI runtime.

“In scope for OCI base layers” means - the TDC and TOB should discuss and make final determination, but these items are part of the initial work for OCI related to container format and runtime

“In scope for OCI optional layers” means-the TDC and TOB should discuss and make final determination, but these would be good examples of items to be discussed under the OCI organization as optional (but well defined) layers on top of container format or runtime. By incorporating these optional layers, we also help ensure that the base layers are compatible with a variety of more complete use cases.

“Out of Scope” means that this topic does not belong in OCI, but could be discussed in other foundations or venues.

### TABLE OF ITEMS

What	In/Out/Future	Status	Description	Why
Runtime executable reference	In scope for OCI base layer	In progress	RunC A reference implementation, but not the only implementation that can be compliant with the runtime spec	A runtime helps to ensure that the spec stays usable by consumers and provides a default implementation of the spec.
Runtime Spec	In scope for OCI base layer	In progress	Parameters of the environment needed to run. Should enable the creation of multiple different compliant runtime implementations	The specification contains the inputs to a container so that a compliant runtime is able to turn a root filesystem into a runnable container.

Bundle Format	In scope for OCI base layer	In progress	How bundle is expressed in the file system before running	Just like the spec, the on disk filesystem layout needs to be standardized so that various runtimes know the on disk structure of the bundle.
Hashing for Content Integrity	In Scope for OCI base layer	Work not yet started	Provide a standardized cryptographic method for ensuring container content has not been altered	This is a generic requirement across almost all use cases to ensure content integrity. However, we will need to deal with the issue of content of bundle changes once run. For example if you have volumes, the hash will only be valid before the first run, unless we agree on hashing only certain immutable elements.
Use of Hash as Content Addressable name for immutable containers	In scope for OCI base layer	Tech team discussing.	Provide a mechanism for using the hash as a name	Using a hash as a name is a way to ensure a unique and container name without relying on a particular naming authority/or system. Using hashing for name is an acceptable addition as it does not encode any centralized namespace.
Archival Format	In scope for OCI base layer	Work not yet started	Description of the serialized format and optimizations of a filesystem bundle.	The specification defines how a filesystem bundle is serialized to ensure interoperability between implementations.

Compliance Test Suite	In Scope for OCI base layer	In progress	Test cases and tools to make sure 1) a reference runtime implementation comply with the Runtime Spec;  2) and a container file-system layout to comply with the Image bundle;	1) allow implementations to achieve compliance with the OCF specification;  2) serve as the testing functions for achieving certification as an OCI Certified Solution;
Specifying way to attach signatures	In scope for OCI optional layer			
Creating Reference spec for optional DNS based naming & distribution	In Scope for OCI optional layer	Not currently being worked	Provide a standardized way to point to a DNS-based unique location for container source, provided, that we also support non-DNS based naming and distribution schemes.	It is reasonable to provide a standardized way to use DNS based distribution in conjunction with OCI without requiring its use. There are many good use cases for DNS based distribution, but not all use cases support this. Furthermore, encoding the location of a bundle into the bundle can cause issues with downloads from alternate locations other than the origin specified in the name. If any code is developed, it should not be part of RunC

Canonical namespace	Out of Scope for OCI	N/A	Enforcing a standardized way to name containers	Different approaches to the intersection of naming and distribution make sense for different environments and are inherently controversial. We should enable, not define. OCI should support multiple different naming & distribution schemes, including DNS-based, current Docker distribution/naming, IPFS, etc.
Standardizing on a particular Distribution method	Out of scope for OCI	N/A		There is no current agreement on how to distribute content, and several different ways to envision it that make sense for different use cases. We want to support multiple federated and non-federated namespaces without imposing a distribution scheme

**MECHANISM FOR ADDING “ROWS” TO THIS TABLE**

The appropriate mechanism for adding, removing or modifying rows to this table (e.g. creating a proposal for an additional optional layer) is to bring it before the TDC. The TOB can be a source of appeal and/or can discuss if there isn’t a clear consensus in the TDC.