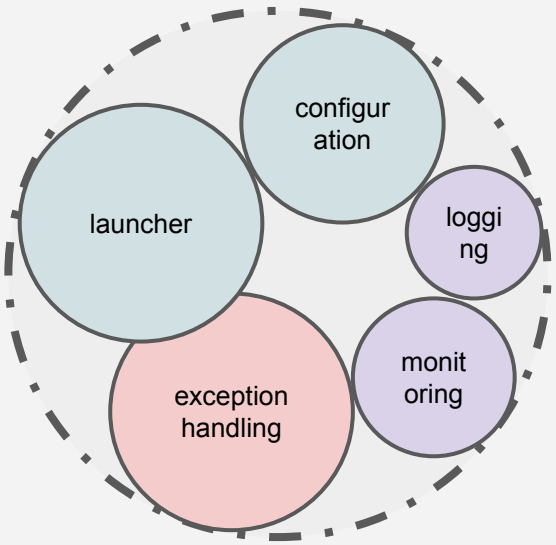
INPUT | OUTPUT

NODE-SHELL



Cardano shell is a thin layer of functionality wrapping the Cardano node. It brings all the other modules working together and makes sure that they have the required resources (configuration, exception handling, monitoring, logging, ...) available.

Integrating in the usual way is tedious (manual editing, maintaining stack.yaml, nix, cabal) whereas using the shell which will act as a single point of integration provides much simpler integration, configuration, initialization and testing/mocking abilities.

Another benefit is that you can provide long-term stable API even if the other internal libraries change their API over time.

Cardano feature

Each feature will define two data types, `CardanoFeatureInit`, and `CardanoFeature`.

```
 data CardanoFeatureInit dependency configuration
layer =
     CardanoFeatureInit
     { featureType :: Text
     , featureInit :: CardanoEnvironment
                   -> dependency
                   -> CardanoConfiguration
                   -> configuration
                   -> IO layer
     , featureCleanup :: layer
                      -> IO ()
     }


 data CardanoFeature = CardanoFeature
     { featureName     :: Text
     , featureStart    :: forall m.
       (MonadIO m, MonadConc m) => m ()
     , featureShutdown :: forall m.
       (MonadIO m, MonadConc m) => m ()
     }
```

CardanoFeature is used to generalize the initialization and the termination for each features.

CardanoFeatureInit defines how the feature can be initialized as well as cleanup function. Initilization of a feature can produce a layer which we'll talk about it later.

Layer

Layer is a list of functions that each feature produces when initialized.

```
data LoggingLayer = LoggingLayer {
  logDebug
, logInfo
, logNotice
}
```

The other features can then add the layer as a dependency when initializing. Thus resolving the dependencies between the features.

```
createNetworkingFeature
  :: LoggingLayer <- Added logging layer to network
  -> CardanoEnvironment
  -> CardanoConfiguration
  -> IO (NetworkLayer, CardanoFeature)
```

Other responsibilities of the shell:

- Configuration
- Global exception handling
- Resources
- Launcher
- Update system
- NodeIPC