The fact that a StreamId identifies a specific stream, means it can carry information and it can be looked up ('discovered').

In order to identify a stream, Streamingpool uses the concept of StreamId.

In order to identify a stream, Streamingpool uses the concept of StreamId. The fact that a StreamId identifies a specific stream, means it can carry information and it can be parametrized. For example, for accessing the hardware publication of a device, one could create a DeviceStreamId("T1") and then parametrize it with the device identifier, Listing 3:

```
Listing 3: Usage of a hypothetical DeviceStreamId class.
DeviceStreamId streamId = DeviceStreamId.ofDevice("LHC.TUNE.BEAM1");
```

Stream Discovery

The mechanism that allows a user to get streams from the Streamingpool is called DiscoveryService. When initializing the Streamingpool using Spring, a DiscoveryService bean becomes available for injection.

The API of the DiscoveryService consists of a single method: discover(StreamId<T> id) which returns a Publisher<T>.

Usually, in the Streamingpool streams are created lazily, at discovery time. Practically, it means that when the user discovers a StreamId, the Streamingpool triggers its creation if it is not present in the system. Therefore, a call to the method discover(StreamId<T> id) is blocking.

Errors are handled gracefully (i.e. without collapsing any streams). This requires that the involved stream factories are implemented such that they deflect the errors onto a dedicated error stream:

Whenever a stream factory creates a StreamId it returns an ErrorStreamPair of a Publisher<Throwable> for the data and the corresponding Publisher<DeviceData> for the device stream if already present in the Streamingpool makes it easier to build which e.g. can show the relations between the streams or the time structure of the related items. One example of such a generic GUI component which already exists, is a JavaFx panel that shows the exceptions of all error streams provided by a pool, which can be included in any application using Streamingpool as a backend.