Enumeration of data sources

|  |  |
| --- | --- |
| Created by: | Michael Geisinger |
| Reviewed by: | Enter name of reviewer. |
| Status: | Ready for review |
| Last modified: | 2012-03-08 |

|  |  |
| --- | --- |
| Unique identifier: | Enter unique identifier for this scenario. |
| Rationale: | Some use cases require obtaining a list of data sources for a given topic, for example for decision on which exactly to subscribe to or for presentation at a frontend. |
| Example: | An example use case is a room with a set of temperature sensors. Each of the sensors publishes the topic “Temperature” as well as metadata that describe the sensor as well as the quality of the transmitted data (e.g., tolerance, reliability). At a central display screen, the user can select which temperatures to show from a list. This scenario is about defining how the list is obtained. |
| Components: | * DCC * Local Directory * Master Directory * Temperature Sensor Drivers (one on each sensor node) * Display Driver |
| Preconditions: | An unknown set of temperature sensors has been deployed in the room. Every temperature sensor publishes the topic “Temperature” with respective metadata. The temperature sensors as well as the display node are logged into the network. The Display Driver wants to display a list of temperature sensors. |
| Course of events: | 1. The Display Driver calls a DCC API function asking for a list of data sources for the “Temperature” topic and optionally a set of metadata to filter against. In the API call, the Display Driver specifies a callback function to be invoked when a response is available. 2. DCC forwards the request to the Local Directory. 3. The Local Directory forwards the request to the Master Directory. 4. The Master Directory checks for each publisher of the respective topic whether the given metadata to filter against matches the metadata of the respective data source and compiles a list. The list contains the full set of metadata of the found data sources (including unique identifiers for each sending component). 5. The Master Directory forwards the list to the Local Directory. 6. The Local Directory forwards the list to the Display Driver. 7. The Display Driver processes the list (e.g., presents it to the user for selection or determines the set of metadata and/or specific items that should be taken into account). 8. Optional: The Display Driver issues a subscription for a specific set of “Temperature” data sources. |
| Postconditions: | Display Driver knows all data sources for topic “Temperature” at the time of the request. |
| Exceptions: | Concurrent publications for the “Temperature” topic may or may not be included in the list reported by the Master Directory. |
| Limitations: | Formally describe limitations and requirements for the scenario, for example with respect to timing. |
| Test cases: | * List the test cases that prove correctness of the implementation. |
| Design decisions: | * List design decisions that have been made/assumed in this scenario. |
| System states: | * List the system states that are affected by this scenario. |
| Illustration: |  |
| Predecessor scenarios: | * List the scenarios that need to be designed or implemented before this one becomes relevant. |
| Related scenarios: | * List related scenarios. |
| Successor scenarios: | * List the scenarios that require this scenario to be designed or implemented first. |
| Comments and open questions: | * Should the request for the list of data sources be a specific concept in DCC API or should it rather be a “normal” RR mechanism with a dedicated topic? * How should the component be made aware that new data sources are available? For example, if such a request would be issued when the node is not yet logged into the network, the list would only contain local data sources. After subsequent login, a much larger set of data sources might be available, but there is no way to notify the component about that fact. Compare issue #664. |