Distributed Services Architecture (DSA)

DSA brings together entities like devices, services, and applications into one data model that updates in real time. By representing these entities as one system, DSA makes possible or simplifies various tasks, such as analytics, inter-device communication, distributed computing, and application development.

You can read more about DSA on the DSA home page.

Structure of DSA

DSA represents each of the entities in the system as one of the following types:

- **Broker**—A broker does a wide variety of management tasks. For example, a broker manages security, links, subscriptions, and node permissions. Other entities in the system can operate only as permitted by a broker. A broker also saves configuration data to disk. A broker also routes data—that is, it moves data from a source to a destination.
- **DSLink**—A DSLink connects to a broker. A DSLink creates, publishes, and interacts with data. A DSLink can also subscribe to data in the DSA system—that is, it can receive data whenever the data changes. You can find DSLinks in IOT-DSA on Github.
- **Node**—Some nodes are brokers, DSLinks, metrics, actions, or attributes. Other nodes, such as folders, do not fit into any of these categories.
- **Metric**—A metric can exist on any broker, DSLink, or other organizational node. A metric is a key/value pair, in which the value can be any of the data types listed in Supported Data Types, including an arbitrary value map.
- **Data Node**—Data nodes allow data to be stored on the broker's host server.
- **Action**—An action can exist on any broker, DSLink, other node, or metric. An action is an invocable command that can affect an entity. For example, an action might create a node or set a metric value.
- **Attribute**—An attribute can exist on any broker, DSLink, other node, or metric. An attribute is metadata for the selected entity, represented as a key/value pair.

Upstream and Downstream Connections

Upstream and *downstream* connections are important concepts for permissions configuration when you work with multiple servers or with multiple brokers on one server. A downstream entity requests permission, and an upstream entity either grants or refuses that permission. A broker is always upstream from its DSLinks. A broker can be either upstream or downstream from another broker.

Supported Data Types

DSA supports these data types:

- **String**—A sequence of characters or an empty string.
- **Number**—A number or a null value.
- **Bool**—A true or false value.
- **Array**—An array object or a null value. The values in the array are of the dynamic data type. An example array of number

values is `[2,3,5,7,11]`. An example array of map values is `[{"hello":"world","number":1},{"hello":"world"}]`.

- **Map**—A map object containing key/value pairs, or a null value. The key is always a string, and the value is the dynamic data type. An example value is `{"hello":"world","primes":[2,3,5,7,11]}`.
- **Binary**—A byte array expressed as a string, or a null value. The string begins with `\u001Bytes:` and ends with a byte array encoded in base 64.
- **Dynamic**—A value that can be any of the above types.

Installing DSA

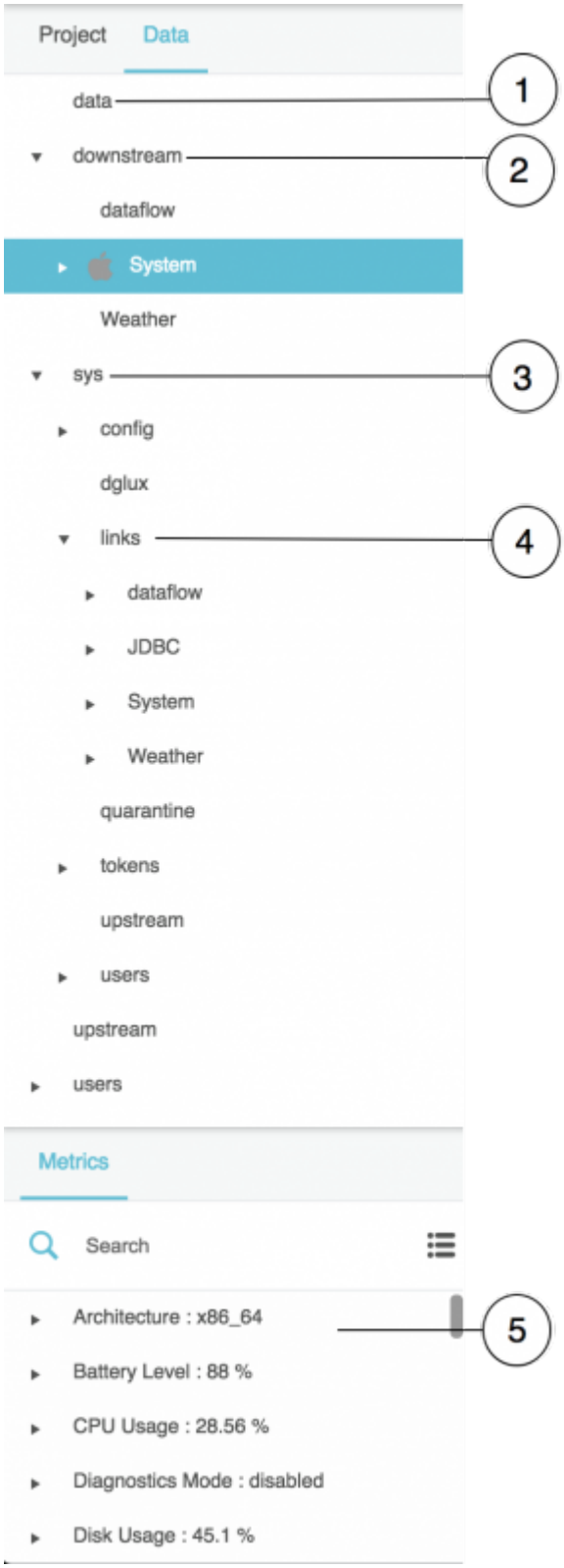
To install DSA on Mac OS X, Microsoft Windows, Linux, or other platforms, follow the steps in the relevant video on this page, depending on your platform.

If you are installing DSA on a Raspberry Pi, BeagleBone Black, or DGBox, you can also follow the steps in this blog post and video.

If you install DSA with DGLux5, and do not have a DGLux5 license, follow the steps to request a license when you are prompted to do so.

Orientation to the DGLux5 Data and Metrics Panels with DSA

The following image shows the Data panel and Metrics panel in DGLux5 when DSA is used.



1	data node	2	downstream node	3	sys node
4	sys > links node	5	metrics at the currently selected node (downstream > System)		

Storing Data on the DSA Server

To store data on the DSA Server:

1. In the Data panel, right-click the data node.
2. Choose **Add** > **Node** or **Add** > **Value**.

   The **Value** option lets you specify a data type.

3. Without moving the mouse pointer away from the pop-up menu, enter a node name, and optionally choose a data type.

   A top-level data node is created.

4. In the Metrics panel, right-click the new data node that you created.
5. Choose **Add** > **Node** or **Add** > **Value** to create a child of this node.
6. Continue adding nodes until all nodes are represented.

To set a value:

1. In the Data panel, select the data node.
2. In the Metrics panel, right-click the metric whose value you want to set.
3. Choose **@set**, enter the value, and click **Invoke**.

To invoke an action on a metric:

1. In the Data panel, select the data node.
2. In the Metrics panel, right-click the metric on which you want to invoke an action.

   A pop-up menu of available data actions is displayed.

3. Choose the action, enter any required information, and click **Invoke**.

Installing and Updating a DSLink

To install a DSLink:

1. In the Data panel, right-click the **sys** > **links** node.
2. Choose **Install Link**, and choose an installation method:
   - If you have a link to a ZIP file, choose **from Url**
   - If you have a ZIP file to upload, choose **from Zip**.
   - If the link can be found in IOT-DSA on Github, choose **from Repository**.
3. Specify a name for the new DSLink.

4. Without moving the mouse pointer away from the pop-up window, click **Invoke**.

   When the link has been successfully installed, a "Success" message appears.

5. To start the link, right-click the link under **sys** > **links** > **[link name]**, and choose **Start Link**.

   When the link has started successfully, it appears under **downstream** > **links** > **[link name]**.

To update a DSLink:

1. In the Data panel, right-click the link under **sys** > **links**.
2. Choose **Update** and choose an update type, and then click **Invoke**.

Remote Dataflow

DSA installations feature a dataflow DSLink, also referred to as *remote dataflow*. This tool is similar to DGLux5 dataflow, described here, also known as *local dataflow*.

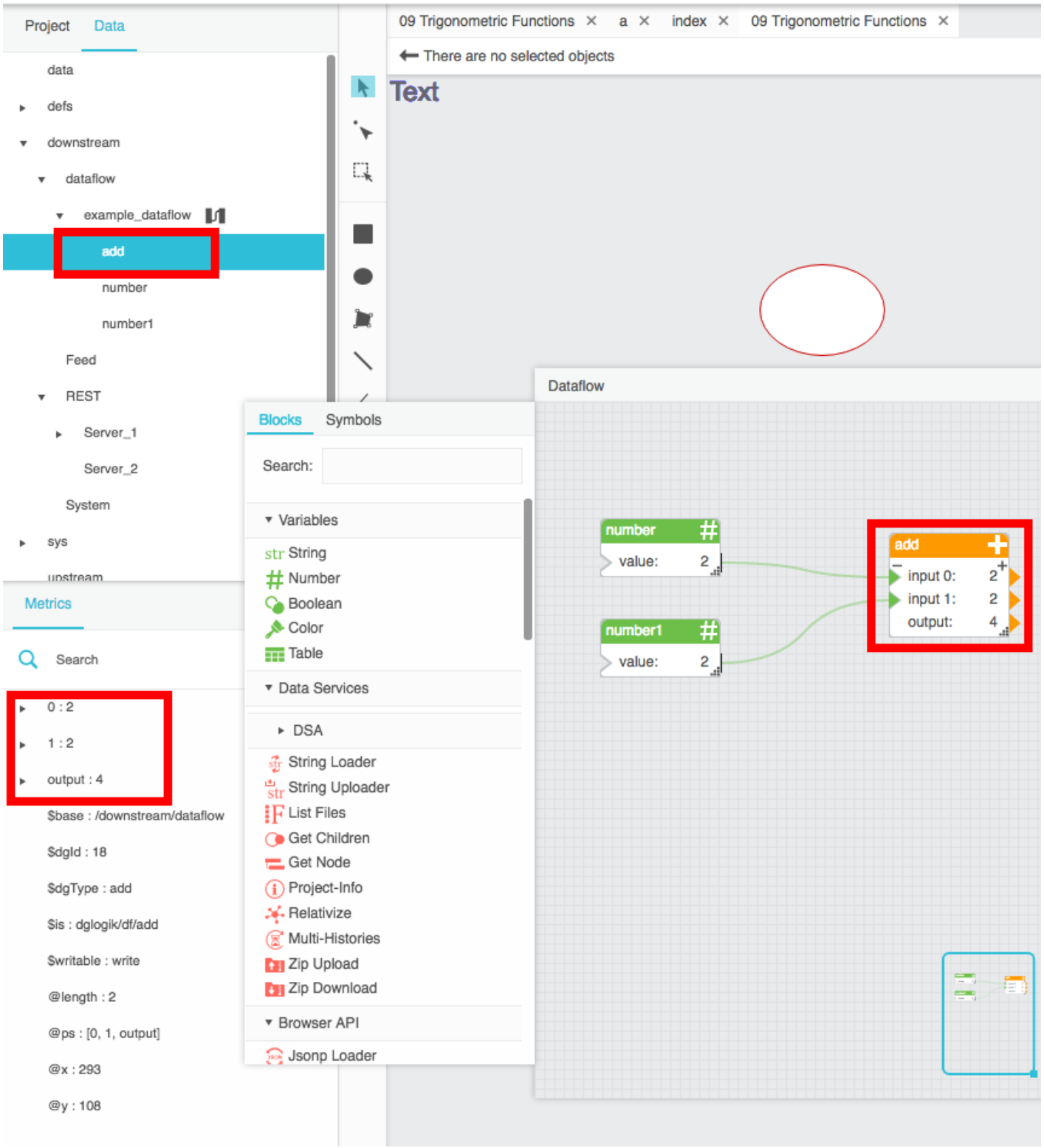
Differences between the two are:

- Remote dataflow is stored on the DSA server with other DSLinks, not in the project files. Therefore, remote dataflow can run even when the project is not open in a browser.
- Certain blocks, such as Browser API blocks, are not available in remote dataflow.

To create a remote dataflow model:

1. In the Data panel, right-click **downstream** > **dataflow**.
2. Choose **Create Dataflow**.
3. Specify a name for the new dataflow model, and click **Invoke**.
4. Choose the new dataflow model at **downstream** > **dataflow** > **[dataflow model name]**.

Block property values in a remote dataflow model appear as metrics in the Metrics panel. They can be used like any other metric. The following image demonstrates an example of remote dataflow metrics.



Security and Permissions

Security and permissions information will go here.