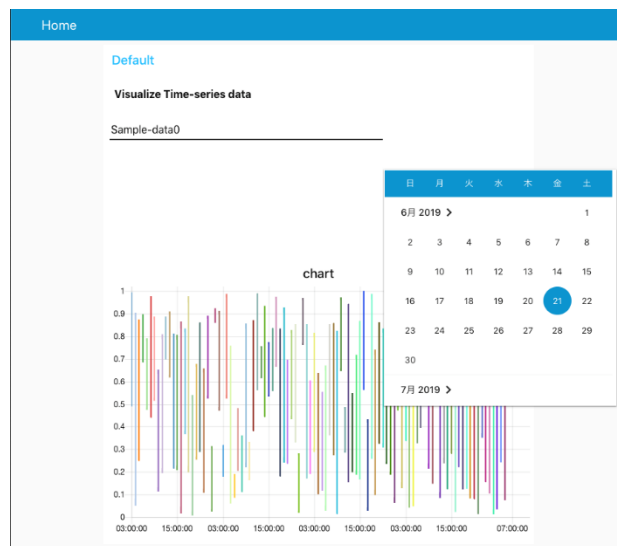




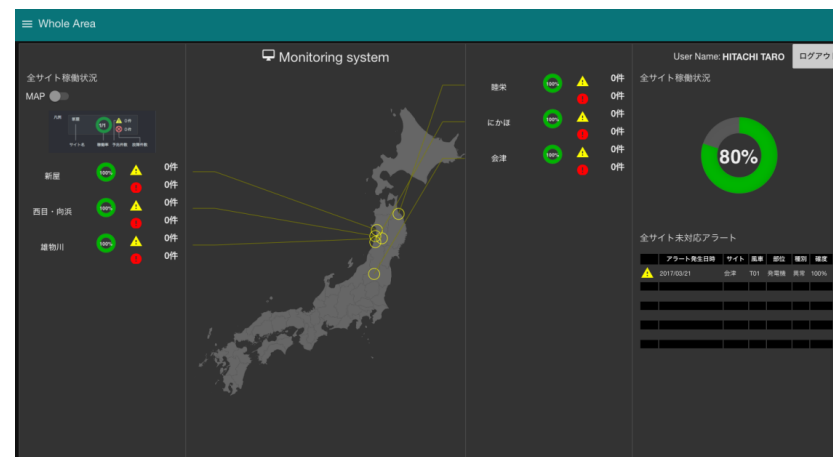
Dashboard Enhancements

Hiroyasu Nishiyama

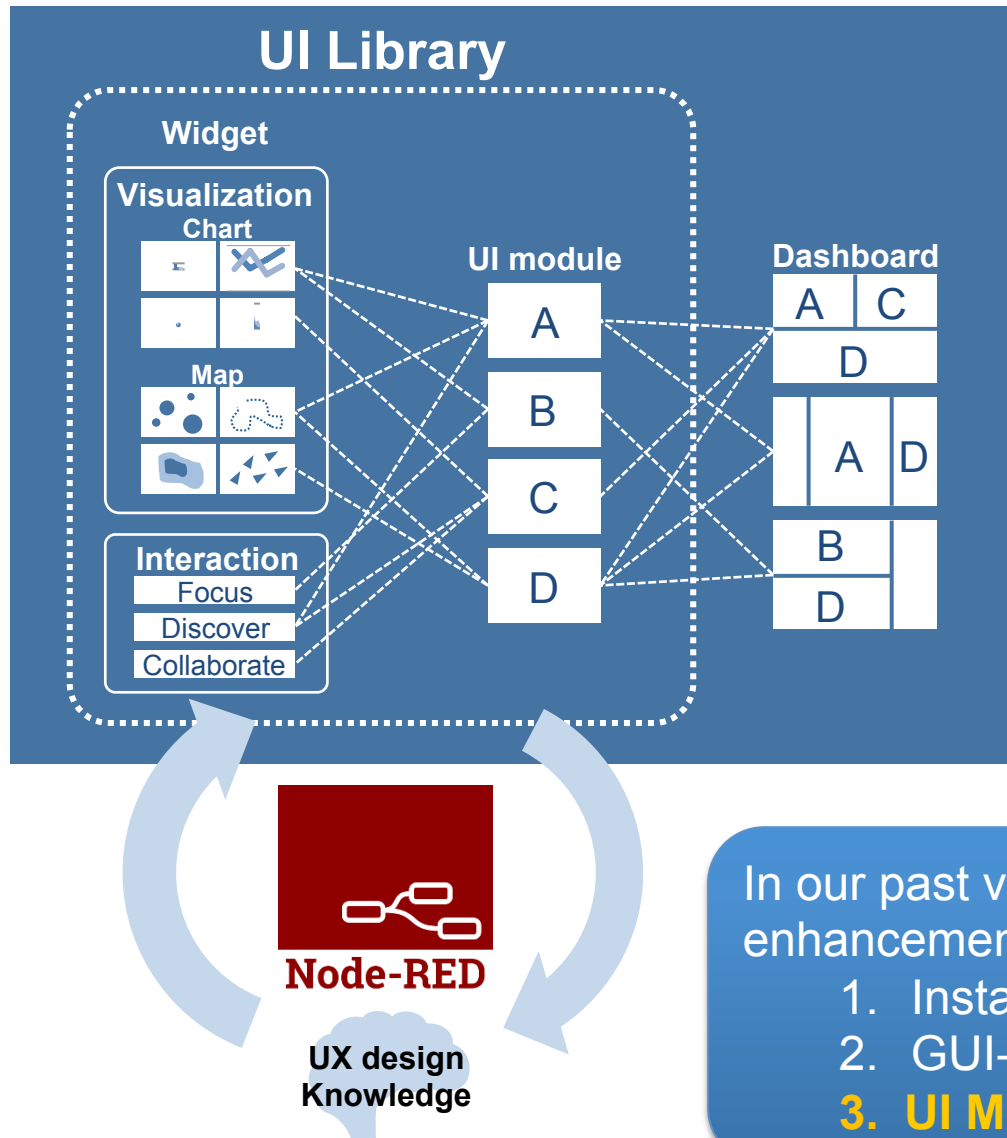
- ❑ Graphical representation of data is effective tool for sharing knowledge among people.
- ❑ Node-RED Dashboard is convenient for creating GUI.
- ❑ But creating complex dashboard is difficult with current Node-RED dashboard.



Simple
Dashboard



Complex
Dashboard



- **Widgets**

Basic design elements that use charts and maps to represent data.

- **UI modules**

Visual components made up of multiple elements.

Each UI module is a group of widgets that meets a particular need in terms of the information it conveys or how it is viewed.

- **Dashboards**

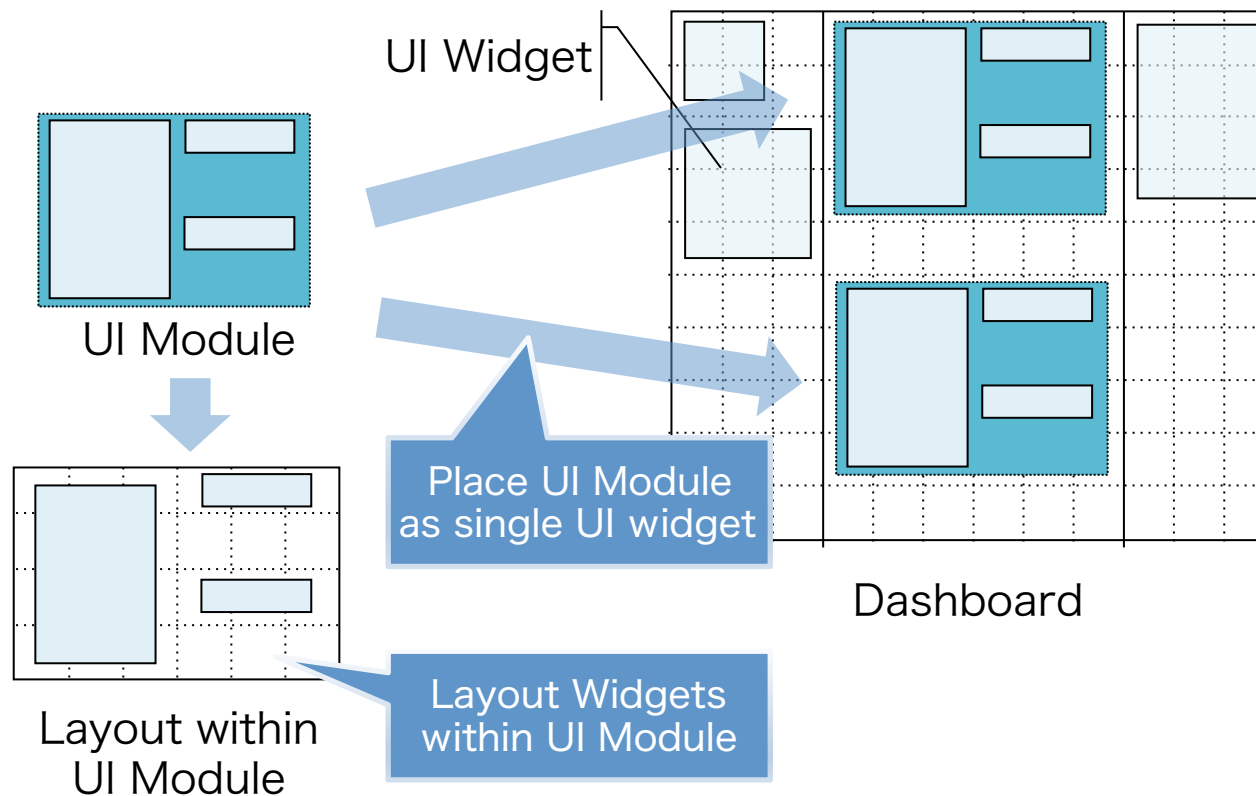
Dashboards that combine a number of UI modules in a predetermined layout.

In our past visit to Hursley, we discussed following enhancements on Node-RED Dashboard:

1. Installable Widgets (design elements)
2. GUI-based Layout Editing
3. **UI Module (Compound Widgets)**

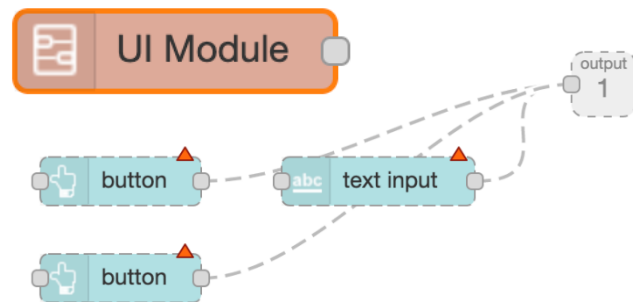
UI Module (Compound Widget)

- ❑ UI module consists of a set of UI widgets.
It has its own internal layout of containing widgets.
- ❑ UI modules can be placed on dashboard similar to UI widgets

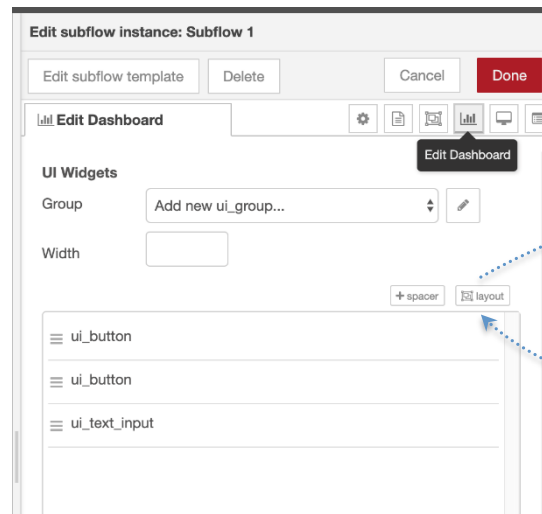


Initial Proposal: SUBFLOW as UI Module

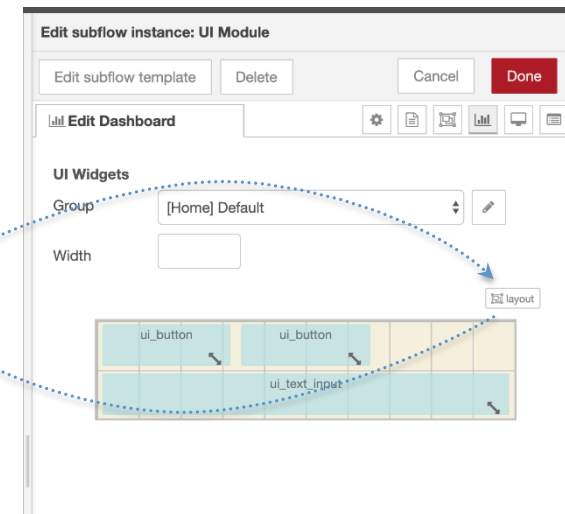
- ❑ UI Module is represented as a SUBFLOW that contains UI widgets
- ❑ SUBFLOW is extended to have interface for UI layout



UI Module (SUBFLOW)
containing 3 widgets



(a) order mode



(b) layout mode

Settings Panel of UI Module

PROBLEM:

Node-RED(SUBFLOW) implementation should be independent of external node module (i.e. Node-RED dashboard).

Discussed solution:
Add feature to edit layout of SUBFLOW UI Module instance to Node-RED Dashboard

The screenshot shows the Node-RED web interface with several panels and annotations:

- Left Panel:** Contains a 'subflows' section with a 'weather' subflow instance and an 'input' section with various modules like inject, catch, status, link, mqtt, http, websocket, tcp, and udp.
- Center Canvas:** Displays a flow with a 'weather' subflow instance (orange box) and a sequence of input modules (button, text input, button) connected by dashed lines.
- Right Panel:** Shows the 'Edit subflow instance: UI Module' dialog box. It includes tabs for 'Edit subflow template', 'Delete', 'Cancel', and 'Done'. The 'Edit subflow template' tab is active, showing a 'UI Widgets' section with a 'Group' dropdown set to '[Home] Default' and a 'Width' input field. Below this is a visual representation of the subflow layout with widgets like 'ui_button' and 'ui_text_input'.
- Annotations:**
 - (1) place SUBFLOW instance: An arrow points from the 'weather' subflow in the left panel to the 'weather' instance on the canvas.
 - (2) push +SUBFLOW button: An arrow points from a '+SUBFLOW' button in the right panel to the 'SUBFLOW' button in the 'Tabs & Links' section.
 - (3) push EDIT button on SUBFLOW: An arrow points from the 'EDIT' button in the 'Tabs & Links' section to the 'Edit subflow instance: UI Module' dialog box.
 - (4) open SUBFLOW edit panel: An arrow points from the 'Edit subflow instance: UI Module' dialog box to the 'UI Widgets' section.
 - (5) set target group and layout of each UI widget: An arrow points from the 'UI Widgets' section to the visual representation of the subflow layout.

Thoughts on the Latest Proposal

PROBLEMS:

- ❑ Can only be applied to SUBFLOW instance (not template)
- ❑ Can not share layout information among exported SUBFLOW

The screenshot shows the Node-RED web interface. On the left, the 'subflows' palette contains a 'weather' subflow. An arrow labeled '(1) place SUBFLOW instance' points to a 'weather' subflow instance placed on the main workspace. A callout box says 'Applied to SUBFLOW instance'. Below this, a sequence of nodes (button, text input, button) is shown with an arrow labeled '(5) set target group and layout of each UI wid' pointing to them. In the center, the 'Edit subflow instance: UI Module' dialog is open. It shows 'UI Widgets' with a 'Group' dropdown set to '[Home] Default' and a 'Width' field. An arrow labeled '(4) open SUBFLOW edit panel' points to the dialog. On the right, the 'dashboard' sidebar is visible. An arrow labeled '(2) push +SUBFLOW button' points to the '+SUBFLOW' button in the 'SUBFLOW' section. Another arrow labeled '(3) push EDIT button on SUBFLOW' points to the 'EDIT' button in the 'SUBFLOW' section. A callout box at the bottom right says 'Difficult to share UI module layout information with exported SUBFLOW'.

(1) place SUBFLOW instance

Applied to SUBFLOW instance

(2) push +SUBFLOW button

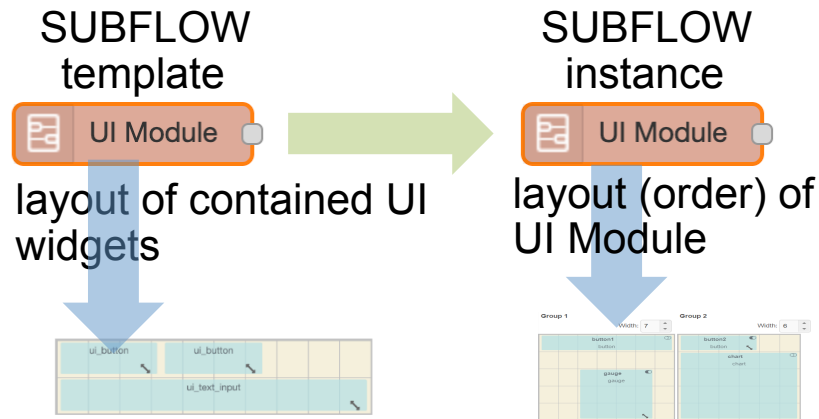
(3) push EDIT button on SUBFLOW

(4) open SUBFLOW edit panel

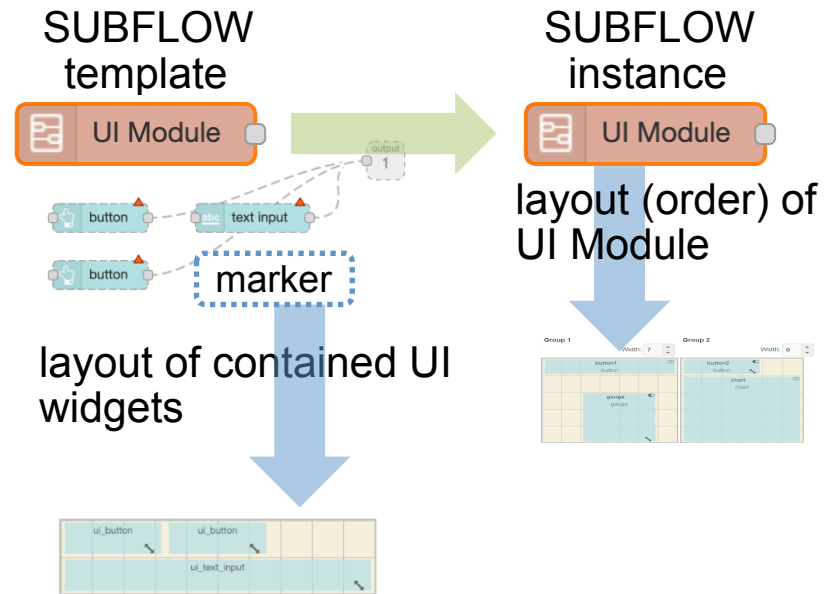
(5) set target group and layout of each UI wid

Difficult to share UI module layout information with exported SUBFLOW

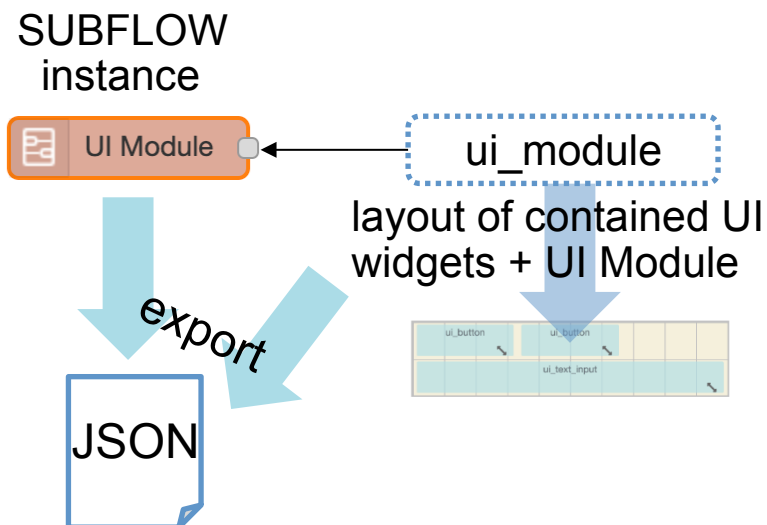
Ideas on Solving UI Module Layout Problem



(a) Extend SUBFLOW



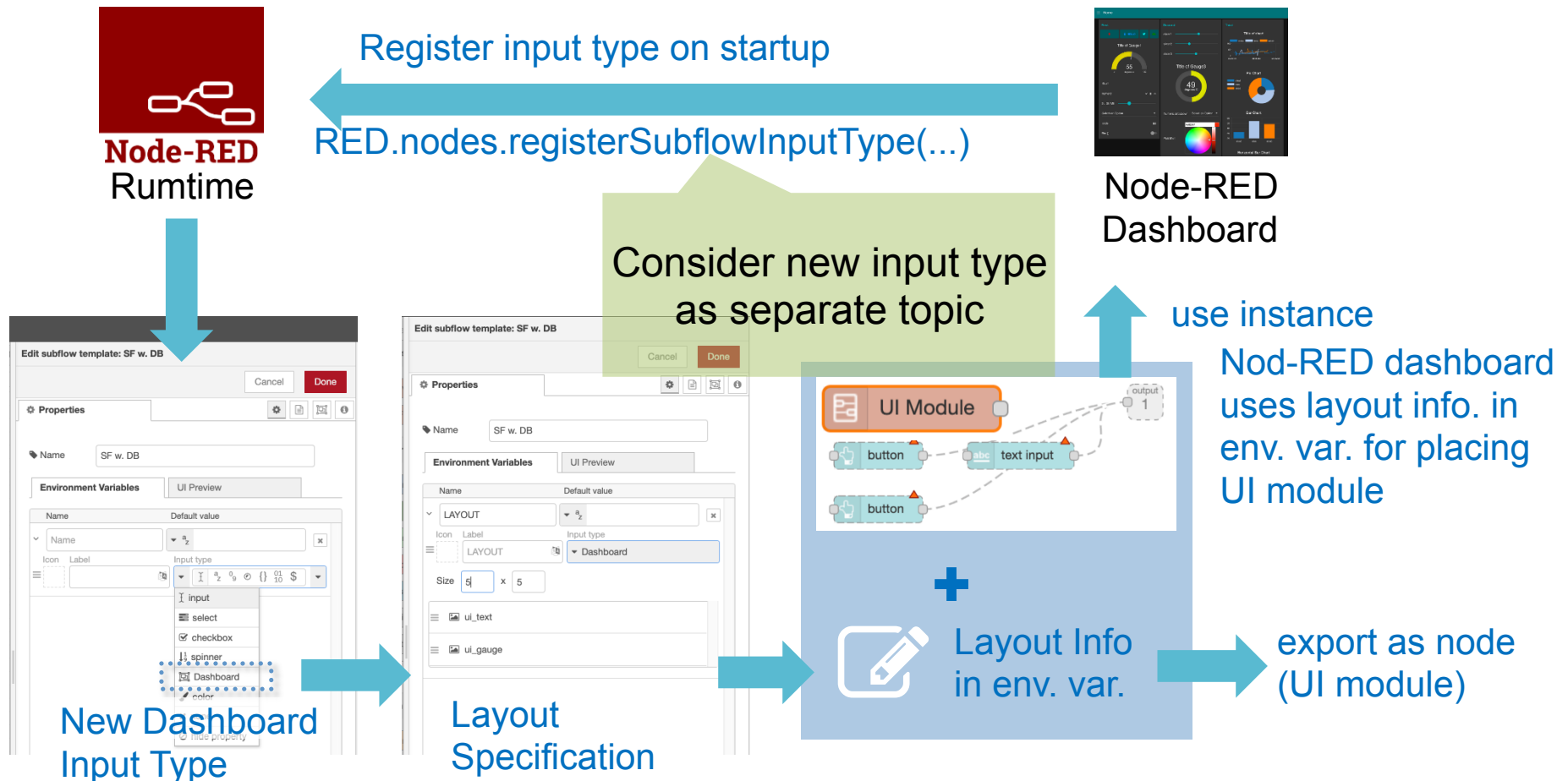
(c) Use Marker Node



(b) Extend Import/Export

New Proposal: Extending SUBFLOW UI

- ❑ Add an API for registering a new SUBFLOW env var type
- ❑ UI for UI module layout information is added using this interface by Node-RED Dashboard.

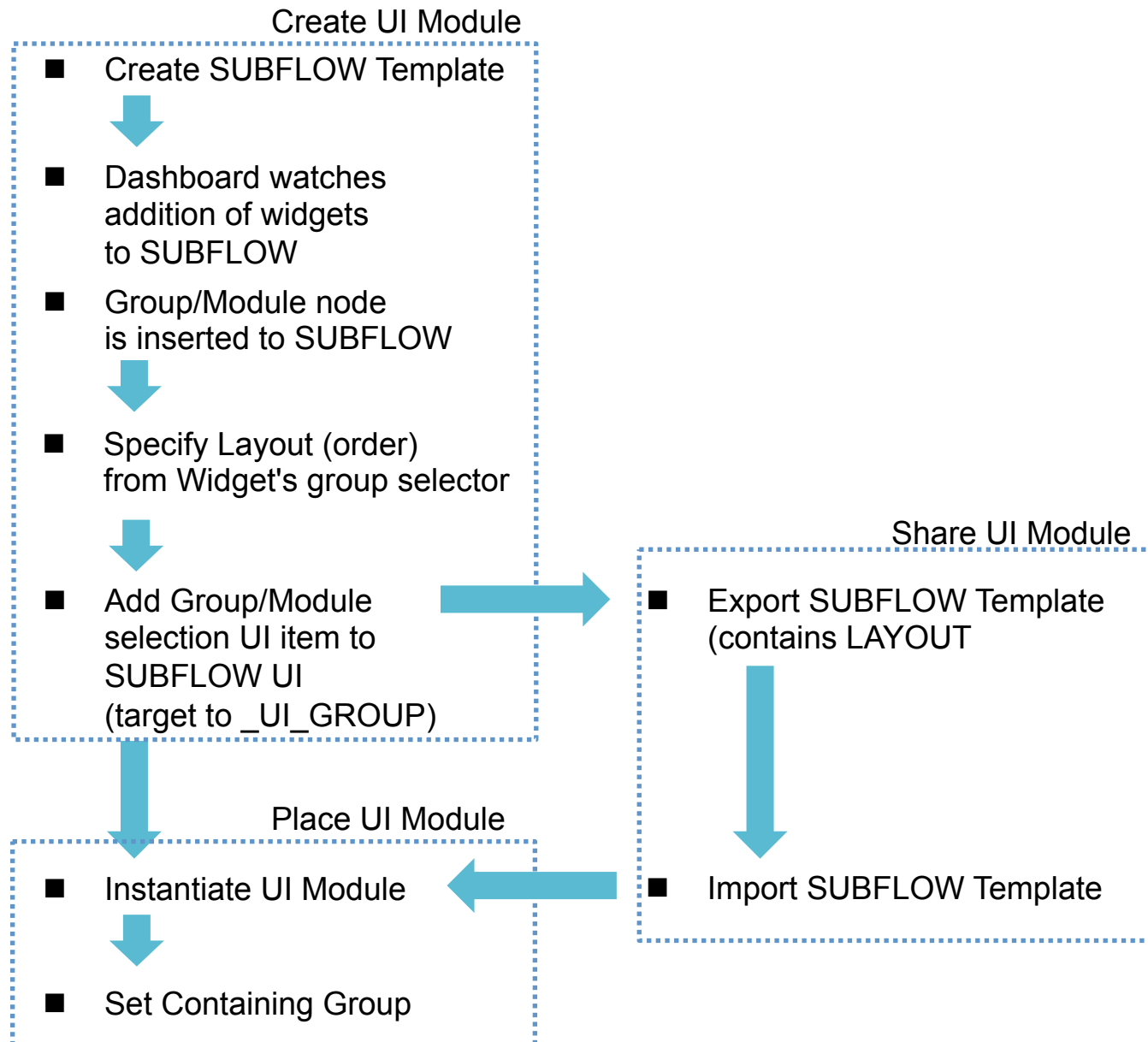


- ❑ Introduce new module config node or extend group node instead
- ❑ Move order information to group node and change it to array of widget node ids
- ❑ Module node is inserted to a subflow when subflow is added to a group
- ❑ config node selection menu type is added to subflow UI
- ❑ Do not allow resizing of module (in 1st version)

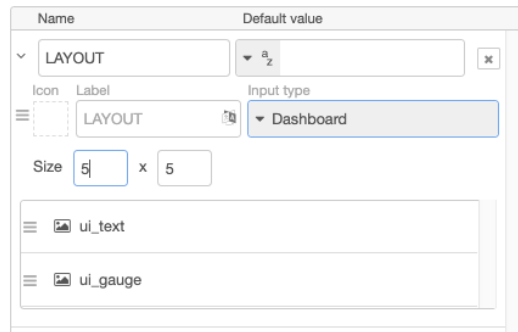
tab

- + group [W1,W2]
 - + W1:widget
 - + W2:widget
- + group [M1, W3]
 - + M1:grup/module[W4,W5]
 - + W4:widget w/h
 - + W5:widget w/h
 - + W3:widget

[N] UI Module Lifecycle

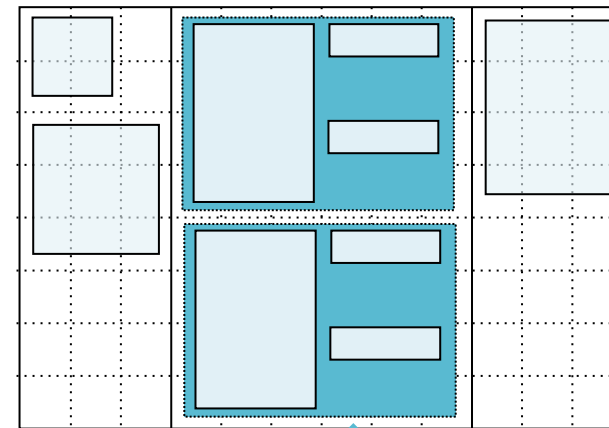


- ❑ UI module (SUBFLOW) must define `UI_LAYOUT` env. var. which contains dashboard layout information of contained UI widgets within SUBFLOW



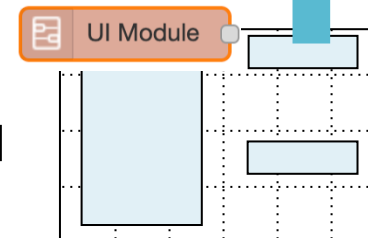
```
{  
  size: [ 6, 4 ],  
  order: [ "<node1>", "<node2>", "<node3>" ]  
}
```

Layout Info in UI_LAYOUT variable



Dashboard

Place UI Module



Layout within
UI Module

User-Defined SUBFLOW Input Type

- ❑ Also useful for various kind of data input
- ❑ Allow selective activation based on SUBFLOW implementation (e.g. activate dashboard input if SUBFLOW contains widgets)

Edit subflow template: SF w.o. DB

Cancel Done

Properties

Name: SF w.o. DB

Environment Variables UI Preview

Name	Default value
COLOR	a_z

Icon Label Input type

COLOR color

(a) COLOR input definition

Edit subflow template: SF w.o. DB

Cancel Done

Properties

Name: SF w.o. DB

Environment Variables UI Preview

COLOR

(b) COLOR input UI

Add HTML5 input types to SUBFLOW UI

Input type

I a_z 0_9 {} 01/10 \$

- input
- select
- checkbox
- spinner
- Dashboard
- color
- none
- hide property

(a) w. dashboard

Input type

I a_z 0_9 {} 01/10 \$

- input
- select
- checkbox
- spinner
- color
- none
- hide property

(b) w.o. dashboard

Reconsider new API if requirements exists

- ❑ registerSubflowInputType API adds new input type for SUBFLOW UI definition
- ❑ Example

```
var createUI = function(row, id, ui) {  
    $("<input/>", { type: "color", id: id }).css({ width: "100px" }).appendTo(row);  
};  
var exportValue = function(input, item) {  
    item.type = "color";  
    item.value = input.val();  
};  
var def = {  
    def: { // Basic info. for UI definition I/F  
        value: "color",  
        label: "color",  
        icon: "fa fa-paint-brush",  
        hasValue: false  
    },  
    createUI: createUI, // Callback for custom UI creation  
    export: exportValue, // Callback for exporting to env. var.  
    isActive: function() { return true; }, // Callback for selective activation  
    onSelection: undefined, // Callback for custom UI definition I/F  
};  
RED.nodes.registerSubflowInputType(def);
```

[N] Adding new input types to SUBFLOW UI

- ❑ Add support of new HTML5 input types (color, date, etc.) to current UI menu item
- ❑ Concerns on registerSubflowInputType API
 1. How to install new input type?
 - ❑ A node that require new input type (e.g. dashboard) registers UI type.
 - ❑ New UI type is only activated if the node exists within a subflow.
 2. Requirements of new input type
 - Many node provides custom input UI (e.g. catch, switch, etc.)

DEMO

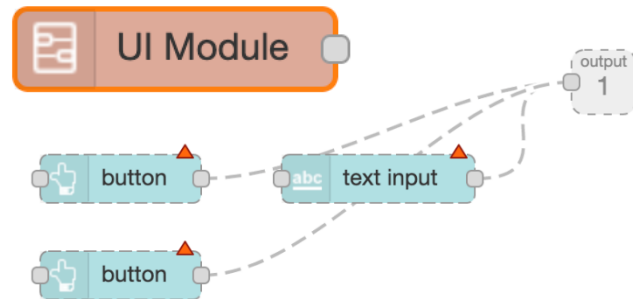
- ❑ Proposed API for adding User-defined Type for SUBFLOW env. var.
- ❑ This allows dashboard layout information for SUBFLOW UI module can be specified using SUBFLOW UI
- ❑ New API is also useful for making other data input for SUBFLOW UI such as color, date, etc. easy and intuitive.

HITACHI
Inspire the Next 

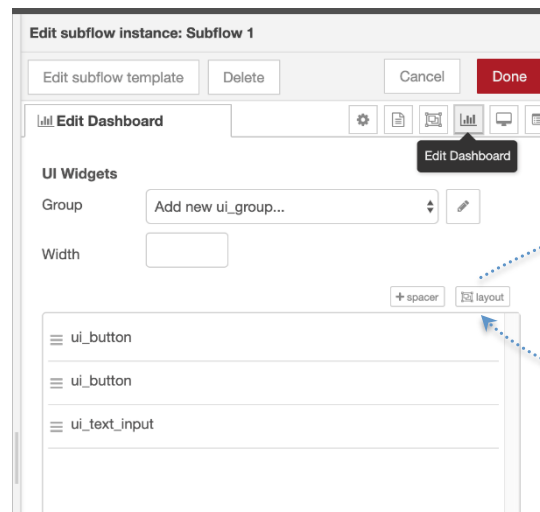
Proposal: SUBFLOW as UI Module

- ❑ Since UI Module consists of a set of nodes, using SUBFLOW as UI Module is natural extension
- ❑ Add interface to specify internal layout of SUBFLOW

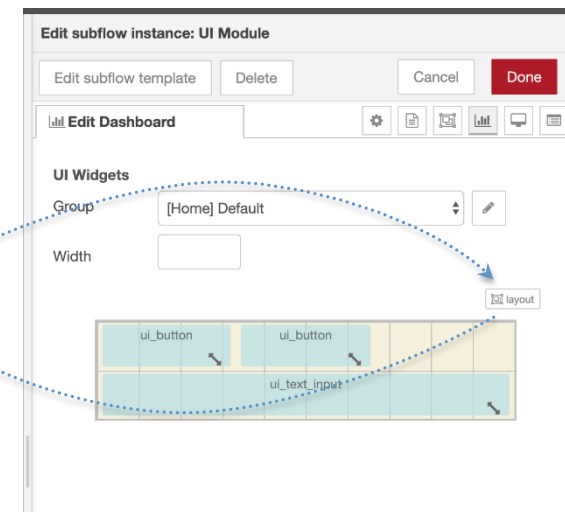
- UI Module (SUBFLOW) has "Edit Dashboard" Tab in settings panel.
- It can specify group, width, and layout.
- Layout can be switched between order-based layout and GUI-based layout (toggle by layout button).



UI Module (SUBFLOW)
containing 3 widgets



(a) order mode

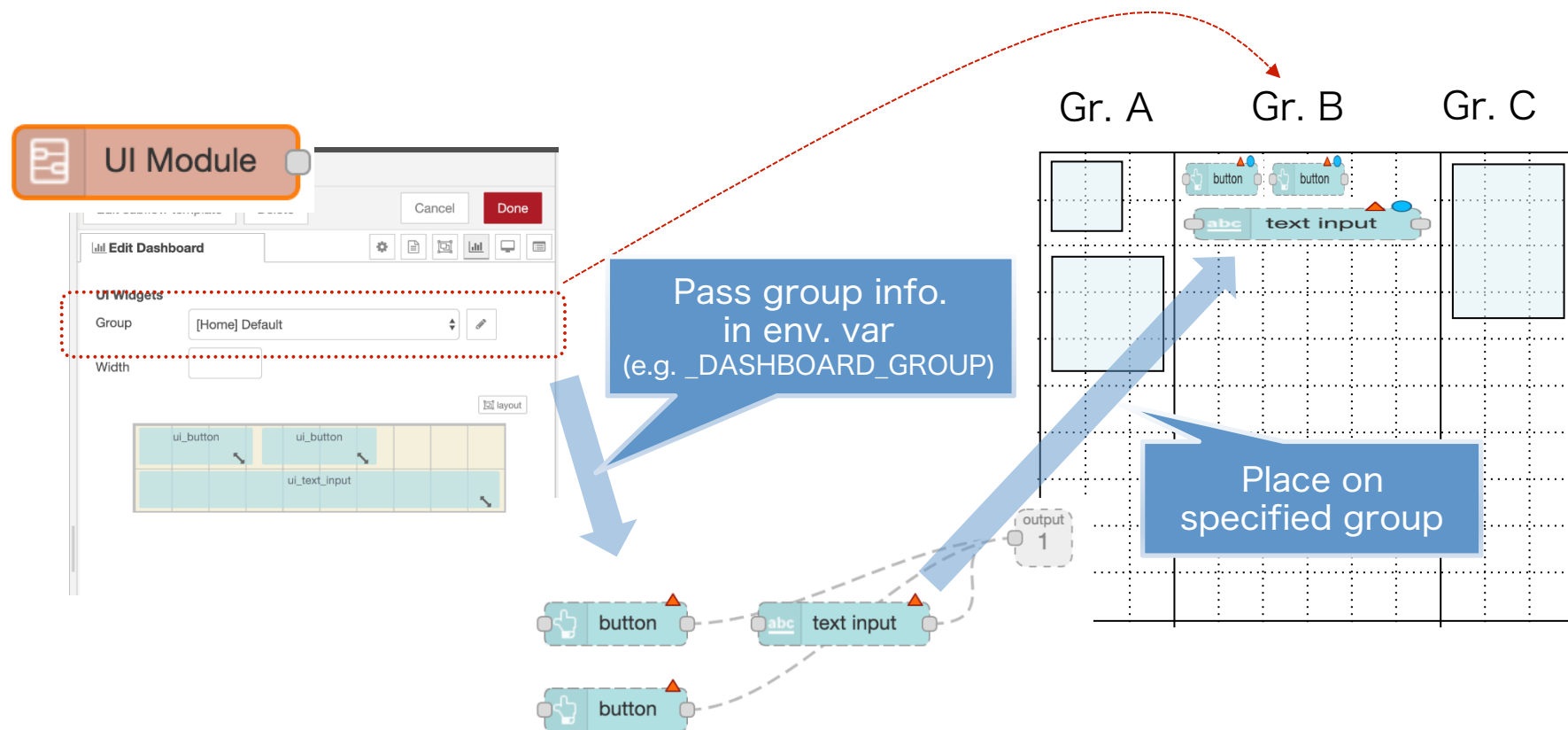


(b) layout mode

Settings Panel of UI Module

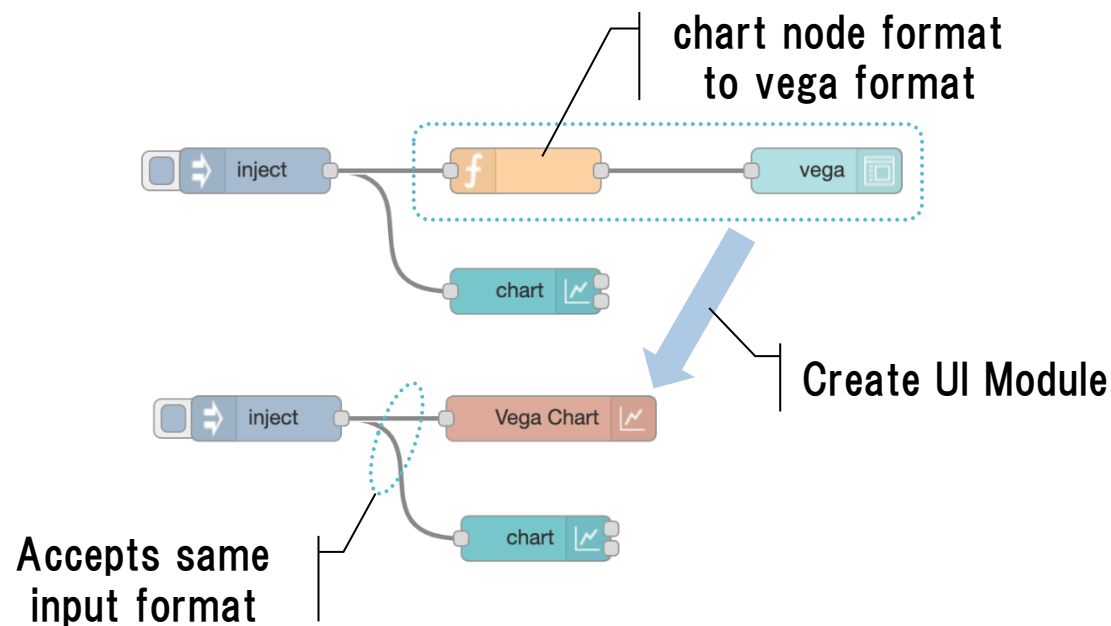
Layout in Dashboard of UI Module

- ❑ UI Module appear as a widget in dashboard layout tab & tool.
- ❑ Group specification of widgets within UI Module is ignored but UI Module's group is passed by environment variable.



Example Usage: Specializing Vega Node

- By using SUBFLOW as UI Module, we can include some logic in UI Module
- One example of this usage is specialization of Vega node
 - Vega node accepts complex visualization specification in JSON
 - The JSON specification is sometimes difficult to write
 - Conversion from light-weight format to Vega specification is useful



Specify following properties:

- ❑ Type name, Label, Icon, ...
- ❑ UI for template definition, SUBFLOW env. var. input
- ❑ importing/exporting values

The screenshot shows a configuration window for a subflow input type named 'LAYOUT'. It includes fields for 'Name' (LAYOUT), 'Default value' (a_z), 'Icon' (a menu icon), 'Label' (LAYOUT), and 'Input type' (Dashboard). A dashed blue box highlights the 'Size' field (5 x 5) and a list of UI components: 'ui_text' and 'ui_gauge'. A blue callout arrow points from the text 'UI for template definition' to the dashed box.

UI for template
definition

The screenshot shows the 'Environment Variables' tab. It features a 'COLOR' input field with a color picker icon and a black color box. A dashed blue box highlights this input field. A blue callout arrow points from the text 'UI for SUBFLOW env. var. input' to the dashed box.

UI for SUBFLOW
env. var. input