HITACHI
Inspire the Next

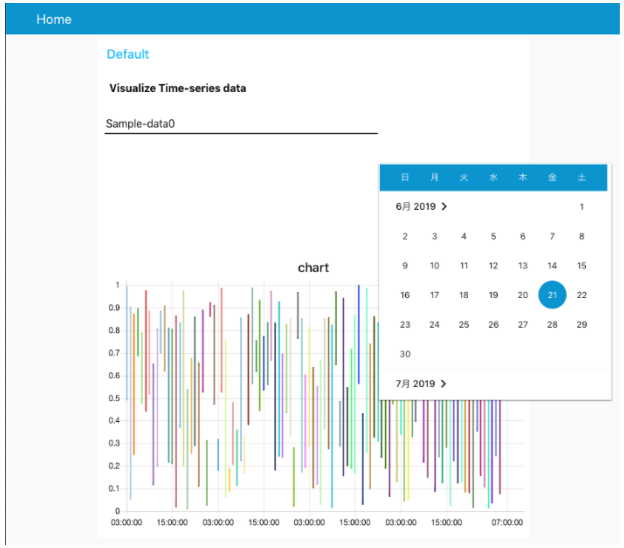
Dashboard Enhancements

Hiroyasu Nishiyama

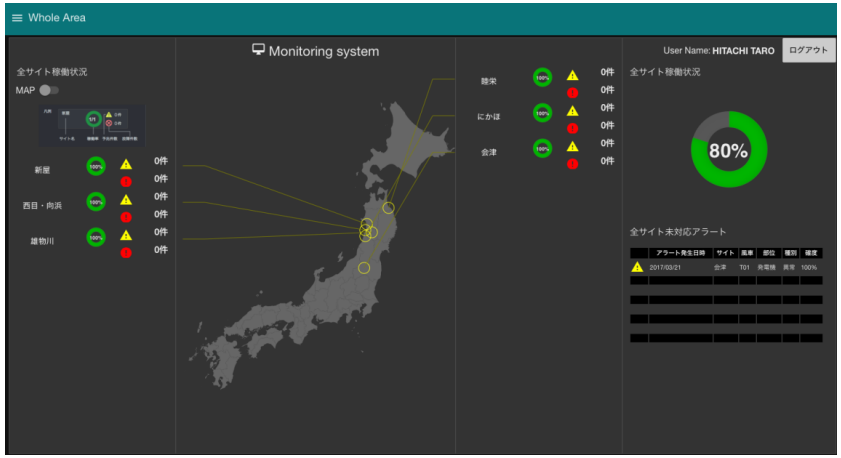
Background

HITACHI
Inspire the Next

- 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.



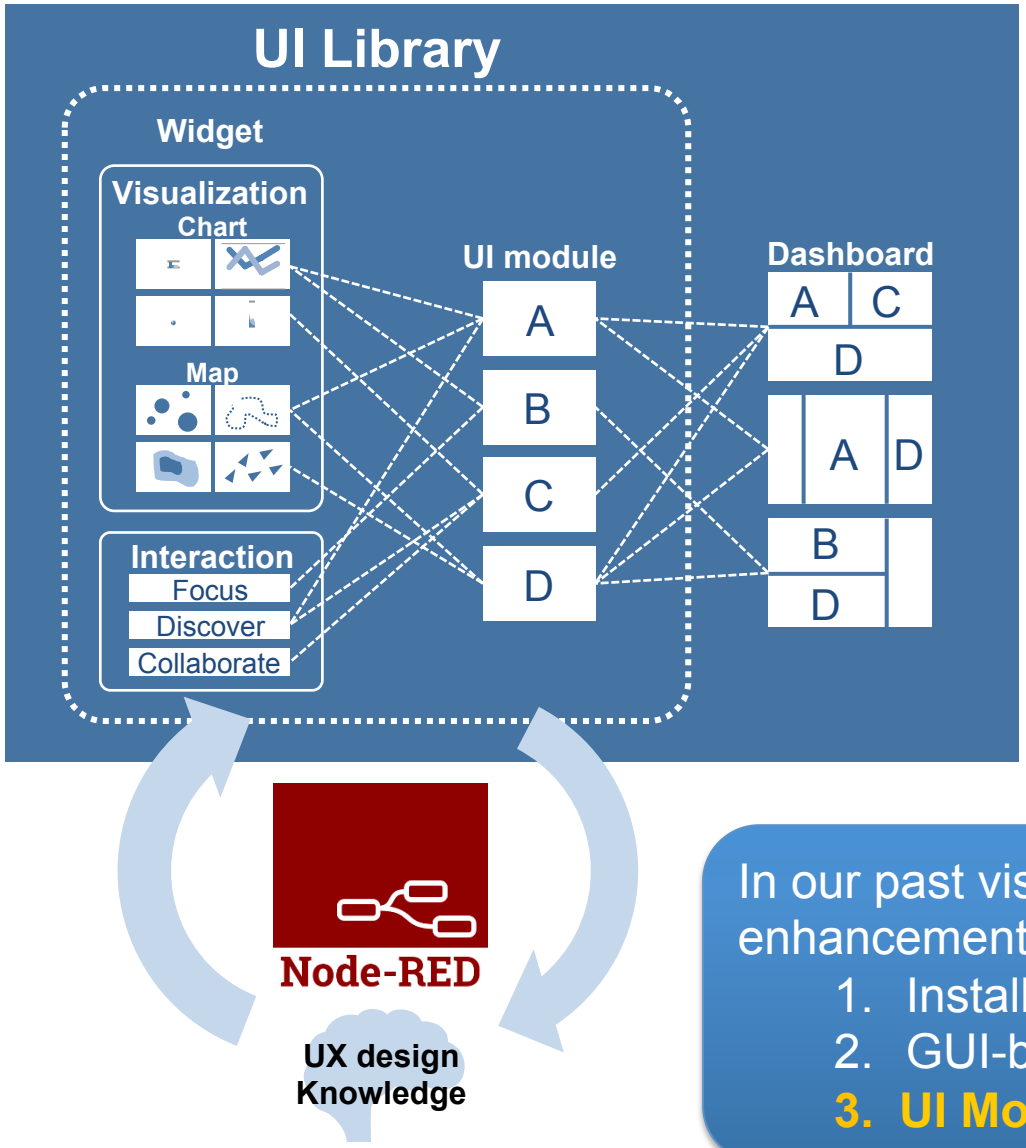
Simple Dashboard



Complex Dashboard

UI Library & Node-RED Dashboard Enhance

HITACHI
Inspire the Next



- **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.

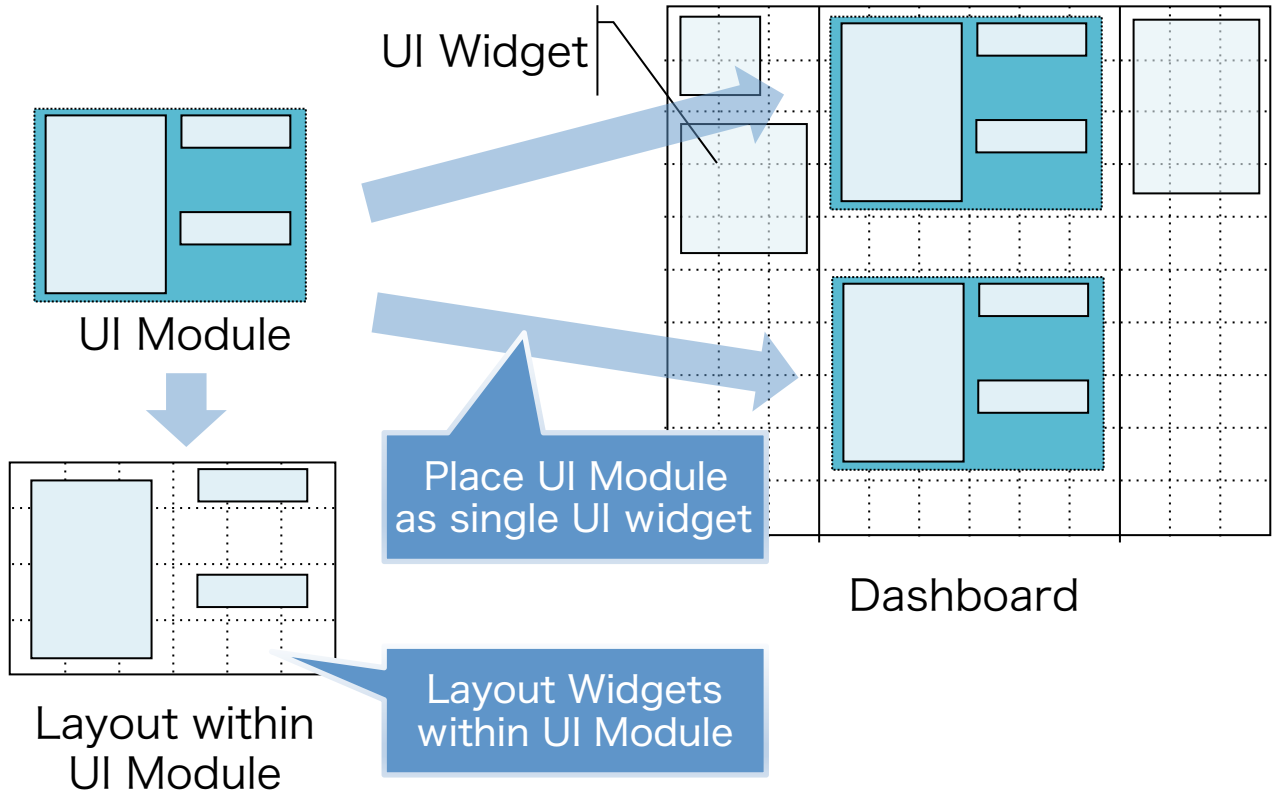
In our past visit to Hursley, we discussed following enhancements on Node-RED Dashboard:

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

UI Module (Compound Widget)

HITACHI
Inspire the Next

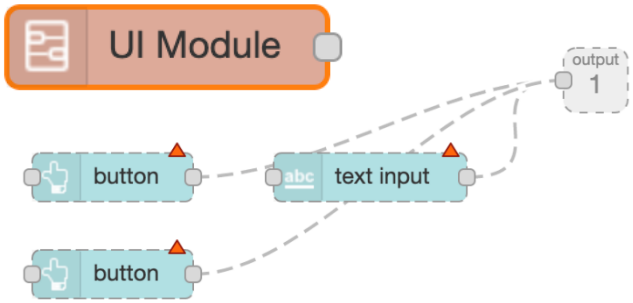
- UI module consists of a set of UI widgets.
  It has its own internal layout of containing widgets.
- UI modules can be placed on dashboard similar to UI widgets



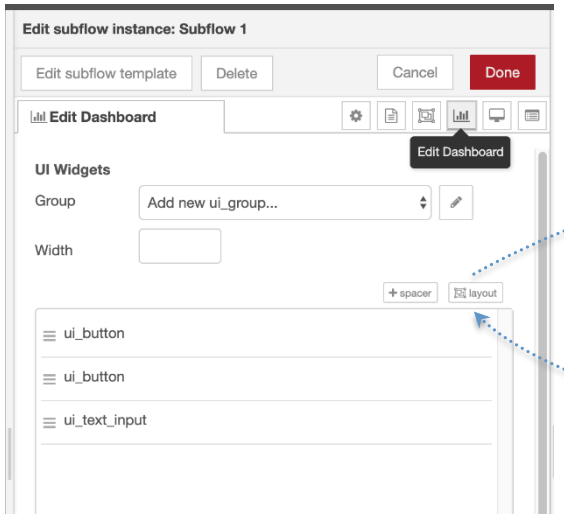
Initial Proposal: SUBFLOW as UI Module

HITACHI
Inspire the Next

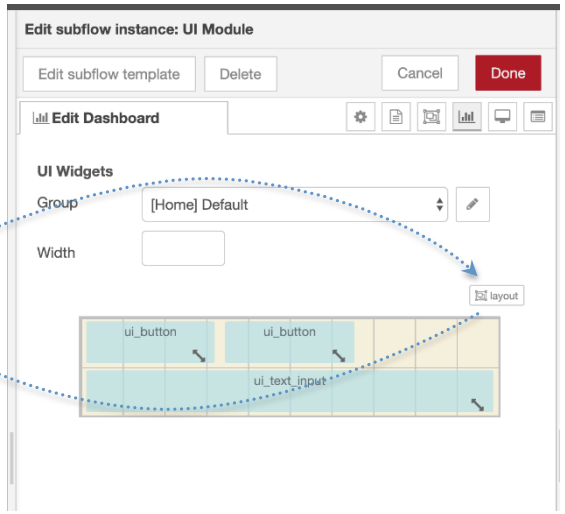
- UI Module is represented as a SUBFLOW that contains UI widgets
- SUBFLOW is extended to have interface for UI layout



UI Module (SUBFLOW) containing 3 widgets



(a) order mode



(b) layout mode

Settings Panel of UI Module

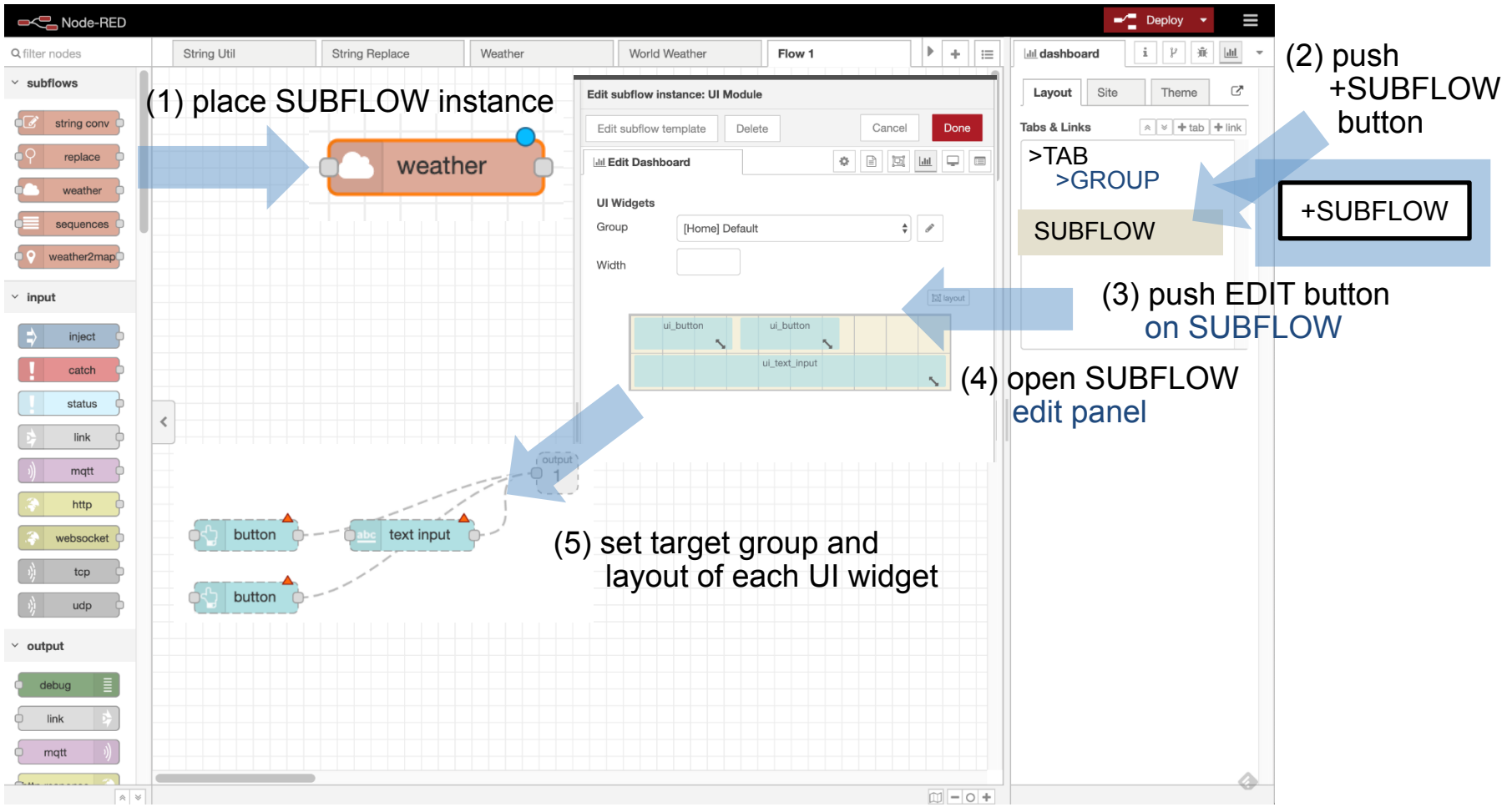
PROBLEM:
Node-RED(SUBFLOW) implementation should be independent of external node module (i.e. Node-RED dashboard).

Discussion on Last Visit to Hursley

HITACHI
Inspire the Next

Discussed solution:

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

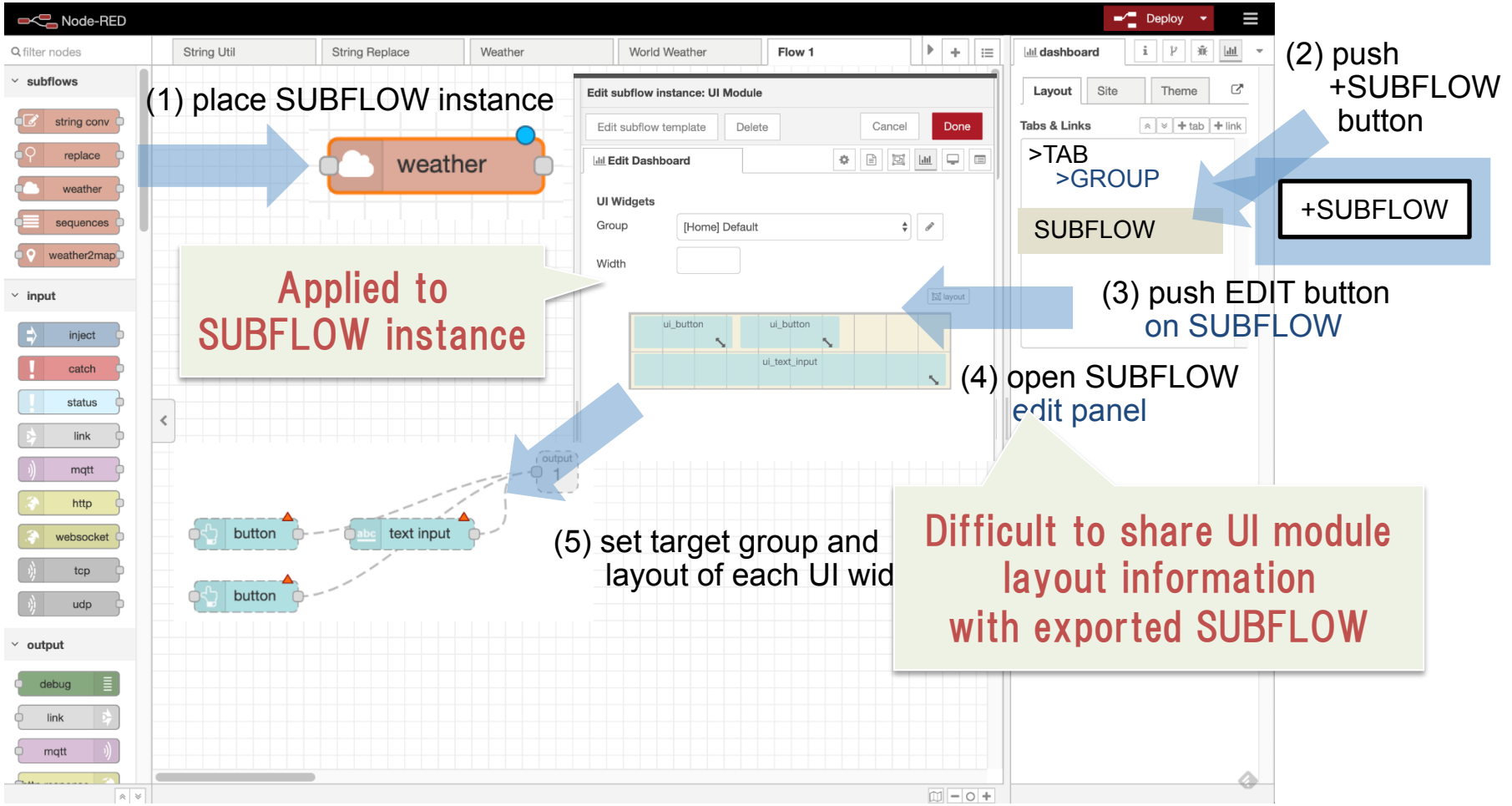


Thoughts on the Latest Proposal

HITACHI
Inspire the Next

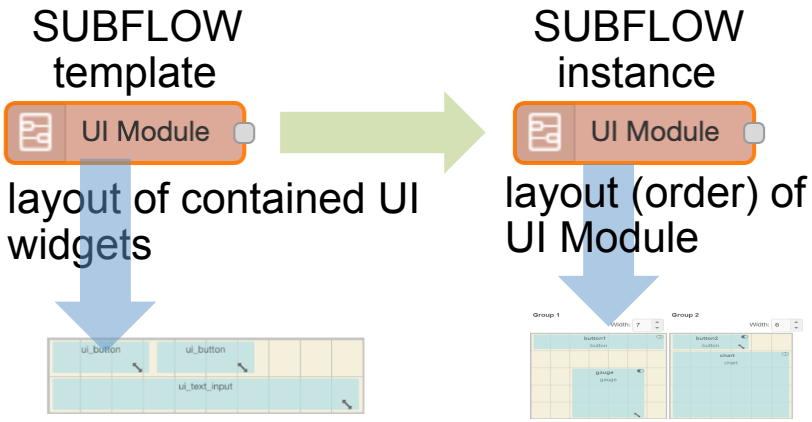
PROBLEMS:

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

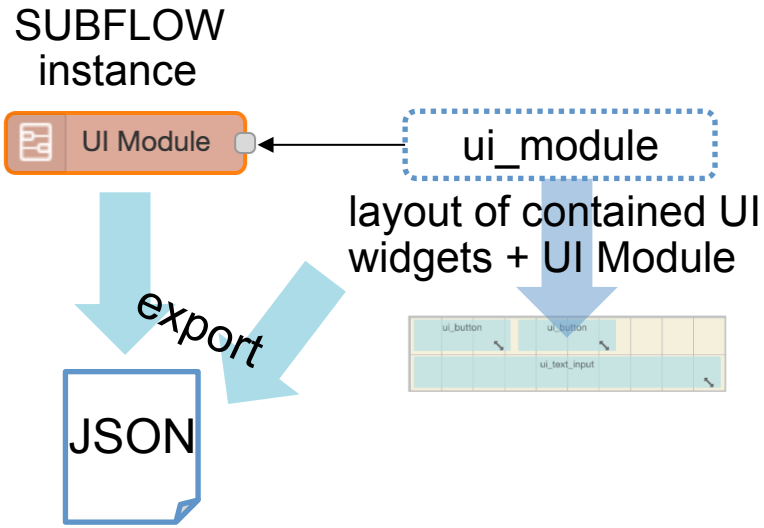


Ideas on Solving UI Module Layout Problem

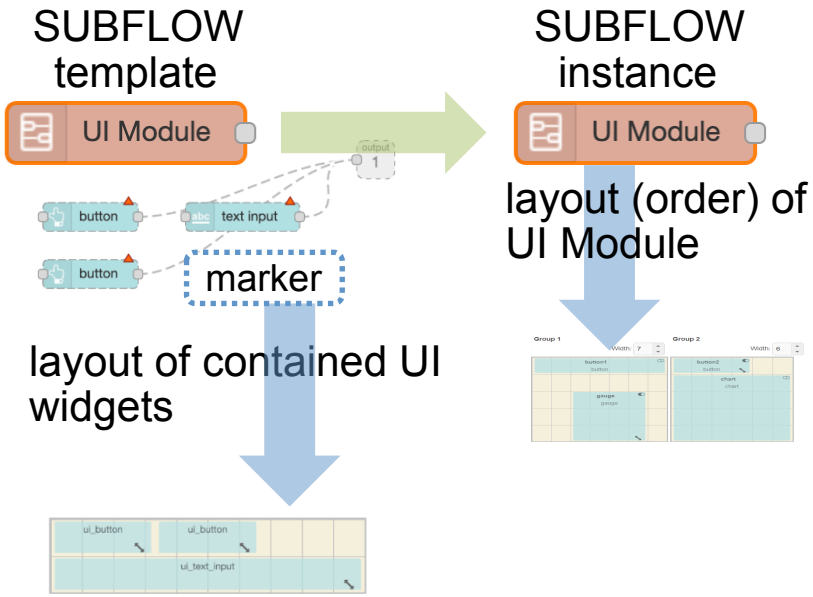
HITACHI
Inspire the Next



(a) Extend SUBFLOW



(b) Extend Import/Export

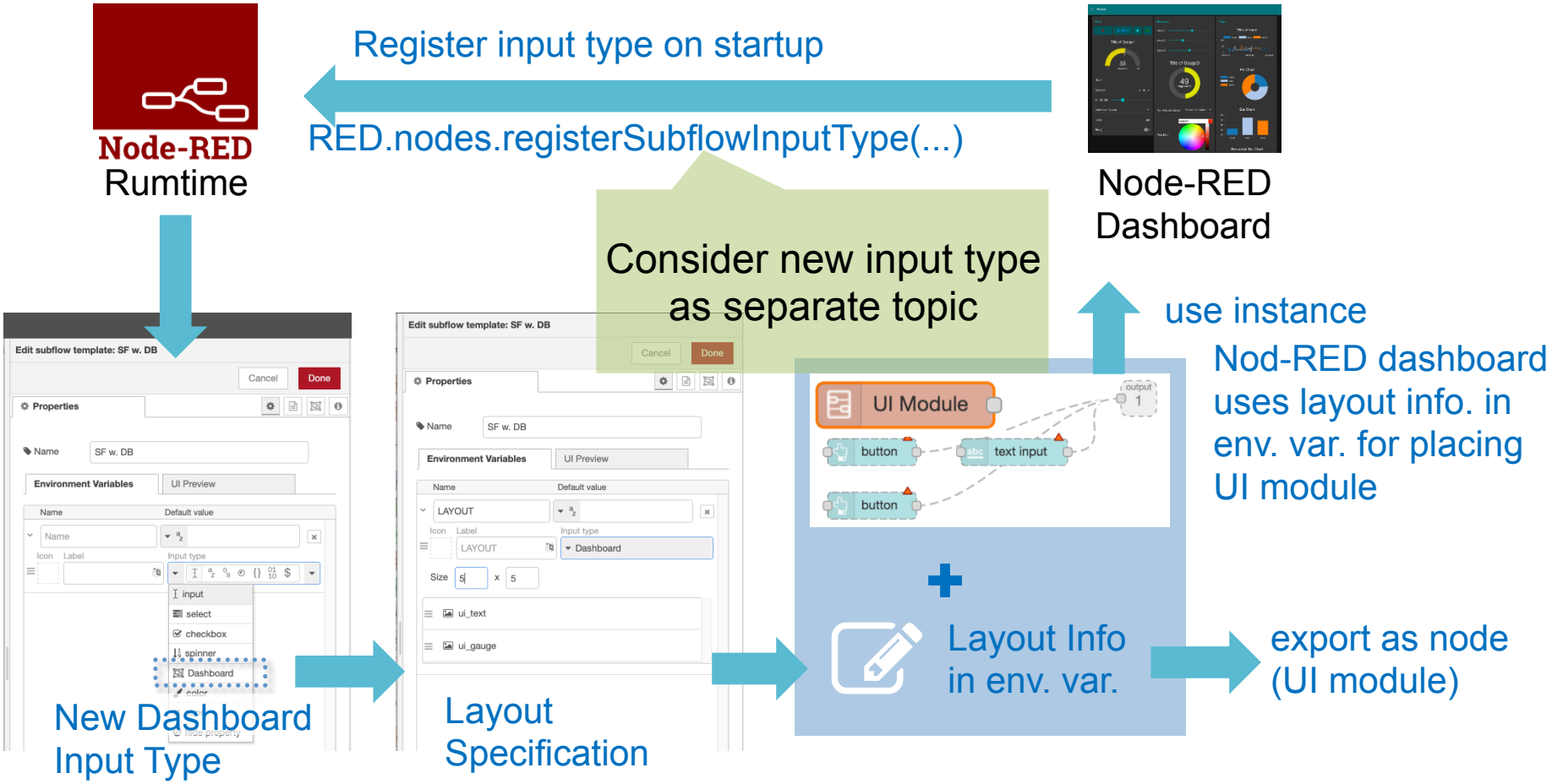


(c) Use Marker Node

New Proposal: Extending SUBFLOW UI

HITACHI
Inspire the Next

- 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.



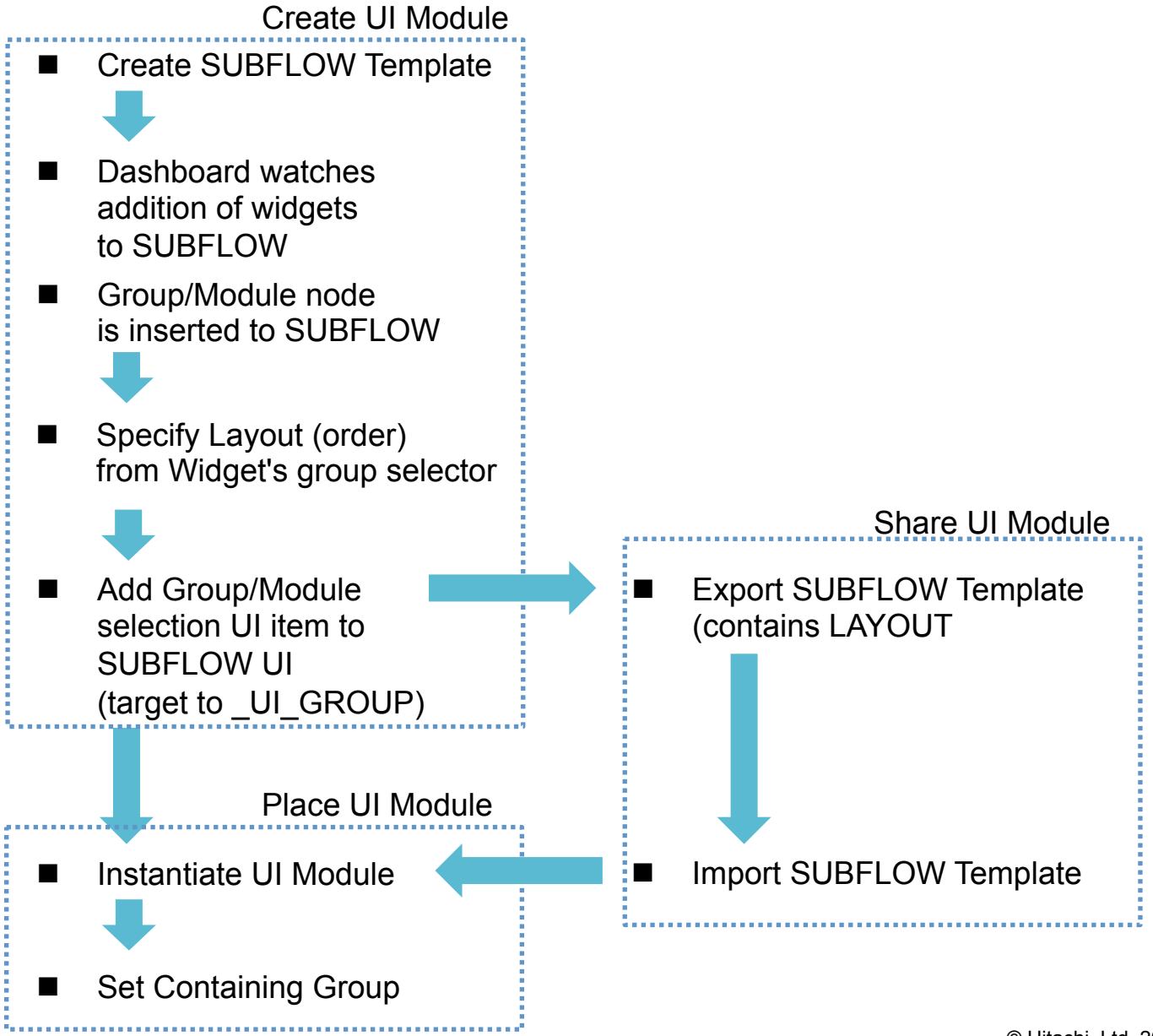
[N]Nick-san's Proposal

HITACHI
Inspire the Next

- 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
```

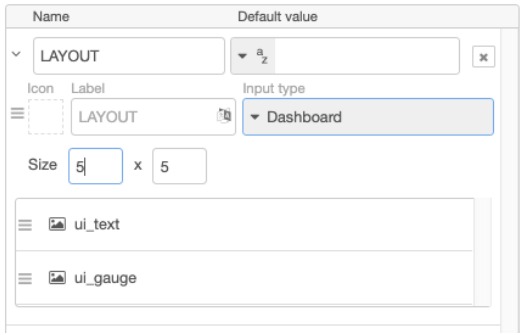
[N] UI Module Lifecycle

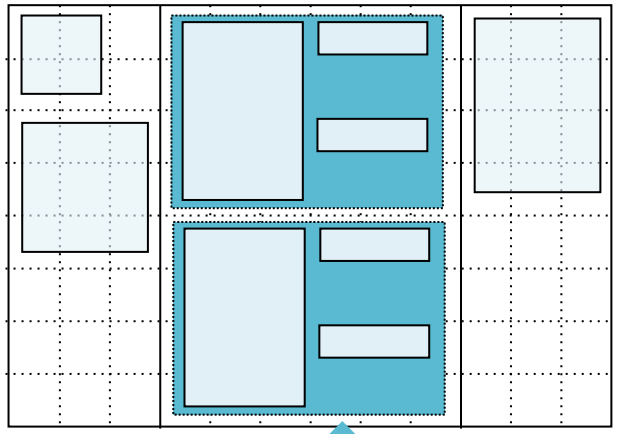


Layouting UI Module

HITACHI
Inspire the Next

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



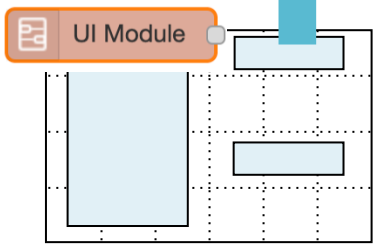


Dashboard

Place UI Module

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

Layout Info in UI_LAYOUT variable

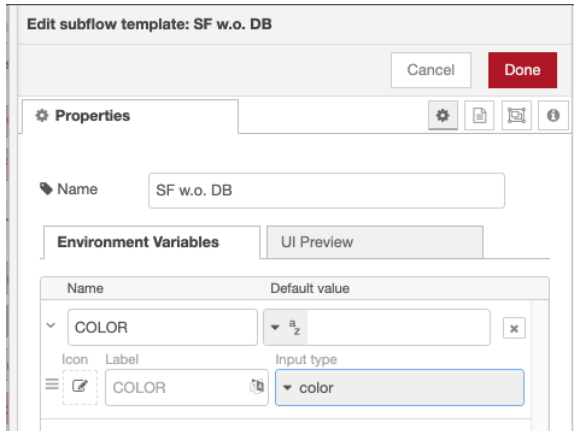


Layout within UI Module

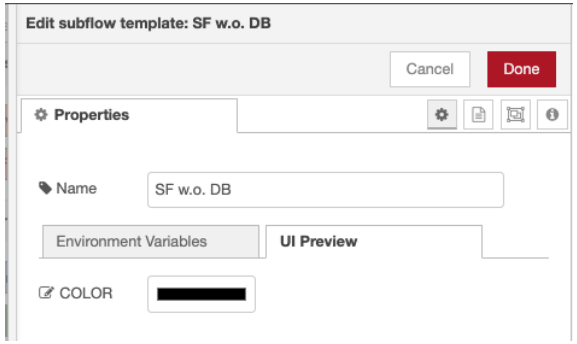
User-Defined SUBFLOW Input Type

HITACHI
Inspire the Next

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

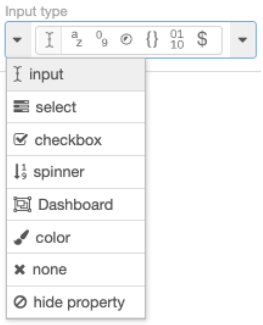


(a) COLOR input definition

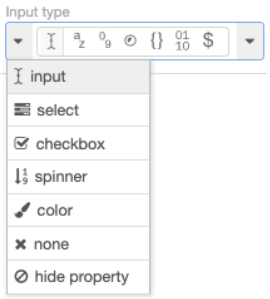


(b) COLOR input UI

Add HTML5 input types to SUBFLOW UI



(a) w. dashboard



(b) w.o. dashboard

Reconsider new API if requirements exists

registerSubflowInputType API[T.B.D.]

HITACHI
Inspire the Next

- 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

HITACHI
Inspire the Next

- 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] User-Defined Input Type

HITACHI
Inspire the Next

DEMO

Summary

HITACHI
Inspire the Next

- 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

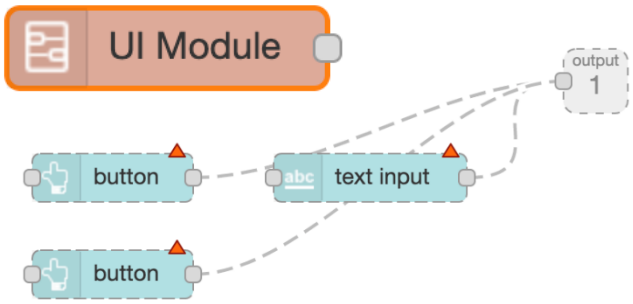
Inspire the Next

Proposal: SUBFLOW as UI Module

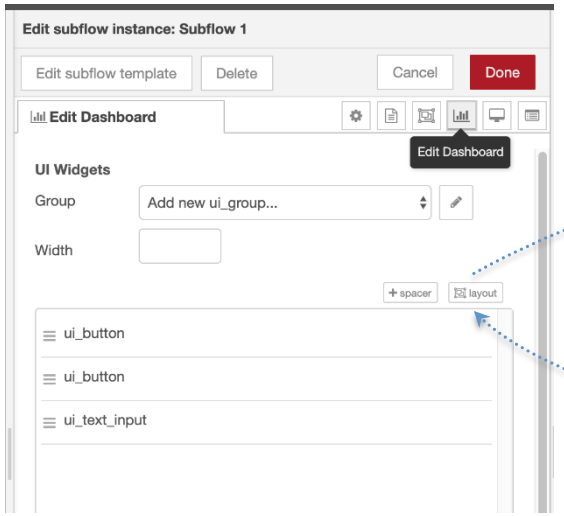
HITACHI
Inspire the Next

- 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

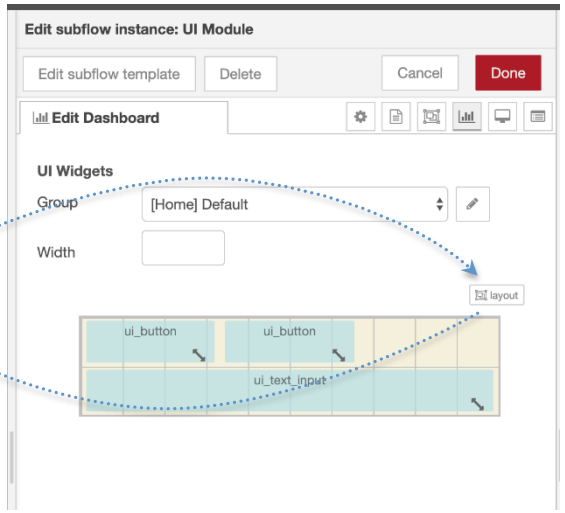
- UI Module (SUBFLOW) has "Edit Dashboard" Tab in settings panel.
- It can specify group, width, and layout.
- Layout can be switched between order-based layout and GUI-based layout (toggle by layout button).



UI Module (SUBFLOW) containing 3 widgets



(a) order mode



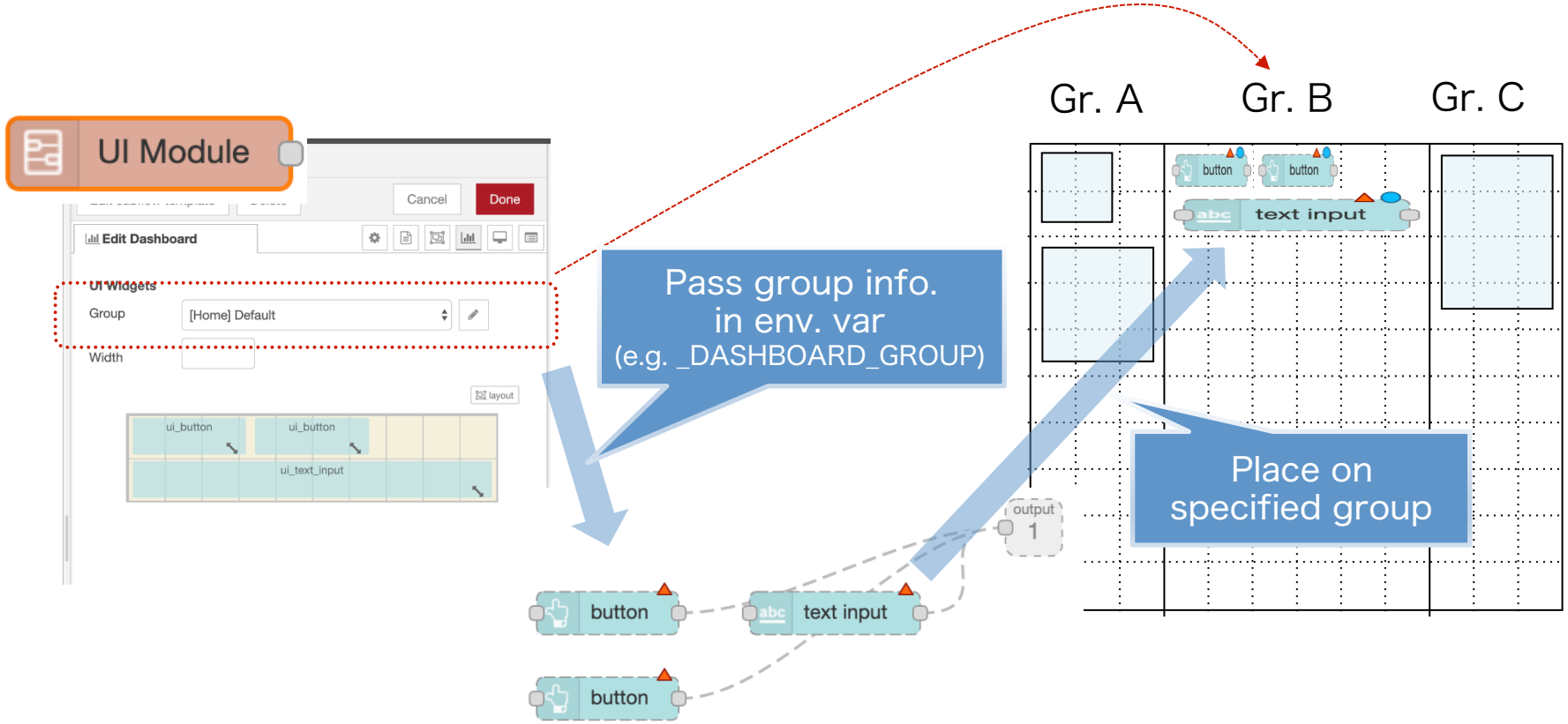
(b) layout mode

Settings Panel of UI Module

Layout in Dashboard of UI Module

HITACHI
Inspire the Next

- 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.

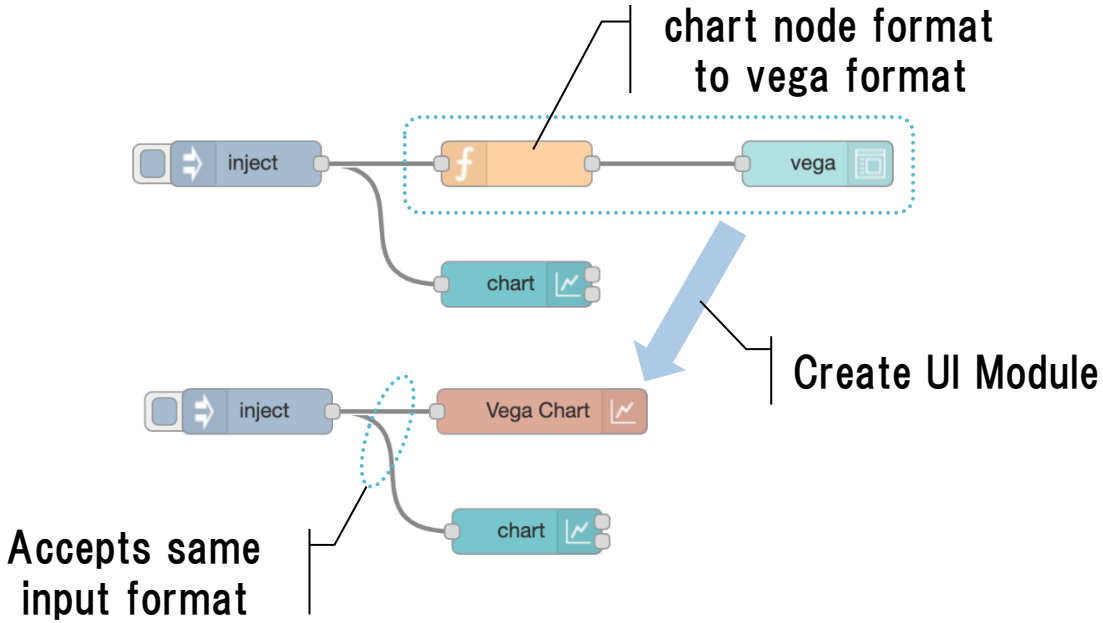


Example Usage: Specializing Vega Node

HITACHI
Inspire the Next

- 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

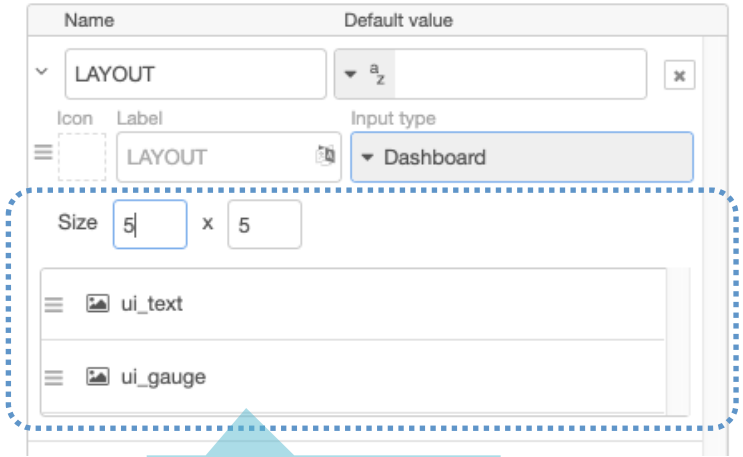


registerSubflowInputType API

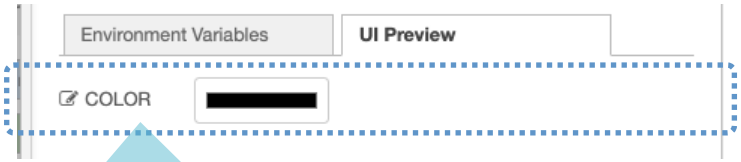
HITACHI
Inspire the Next

Specify following properties:

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



UI for template definition



UI for SUBFLOW env. var. input