

```
(tab-utils/init-tabbed-panel "molecules" :molecules/general)

(utils/init-config-panel "line-chart-demo")

(tab-panel/mk-tab-panel-data data-
  :line-chart/tab-panel.value

(re-frame/dispatch-sync [::events/init-locals :catalog init-db])
```

just call this
:line-chart/tab-panel.value



```
(def app-db
  {:widgets {:<guid-1> {:some-value "value"}
             :catalog {:tab-panel {:value :catalog/atoms
                                   :data-path [:catalog :tab-panel]}}}
   :line-chart {:tab-panel {:value :line-chart/config
                             :data-path [:line-chart :tab-panel]}
    :grid {:include true :strokeDashArray {:d 3 :g 3}
           :stroke "#ffffff"}
    :x-axis {:include true :dataKey ""
             :orientation :bottom :scale "auto"}
    :y-axis {:include true :dataKey "" :orientation
             :bottom :scale "auto"}
    :legend {:include true :layout :horizontal
             :align :center :verticalAlign :bottom}
    :tooltip {:include true}}}})
```

The trick is to generate multiple dependent functions, at run-time, from data

We could make this
:widgets/<guid-1>...

```
(re-frame/reg-sub
 :db/<guid-1>
 (fn [db _]
  (get-in db
   [:widgets :<guid-1>])))
```

```
(re-frame/reg-sub
 :widgets/<guid-1>
 :← [:widgets]
 (fn [widgets _]
  (:<guid-1> widgets)))
```

and we could cascade
this off
:<- [:widgets]...

input-data
context-data

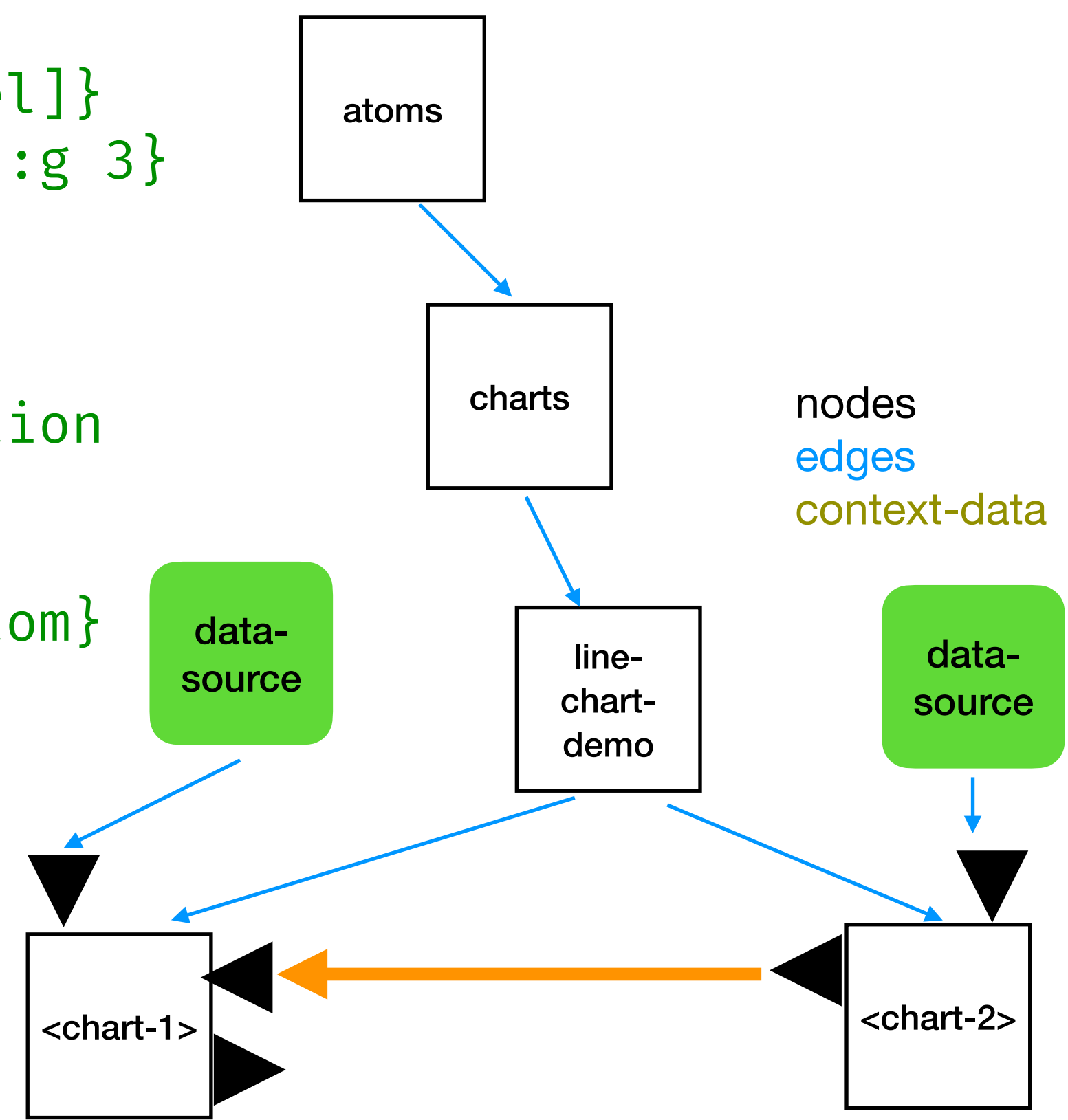
```
(re-frame/reg-sub
 :<guid-1>/some-value
 :← [:widgets/<guid-1>]
 (fn [widget _]
  (get-in widget
   [:some-value])))
```

```
(re-frame/reg-sub
 :<guid-1>/grid.include
 :← [:widgets/<guid-1>]
 (fn [widget _]
  (get-in widget
   [:grid :include])))
```

we could do this as even
more cascaded
subscriptions...

Option 1
Each component is a node
and contains the “edges” to
what composes it

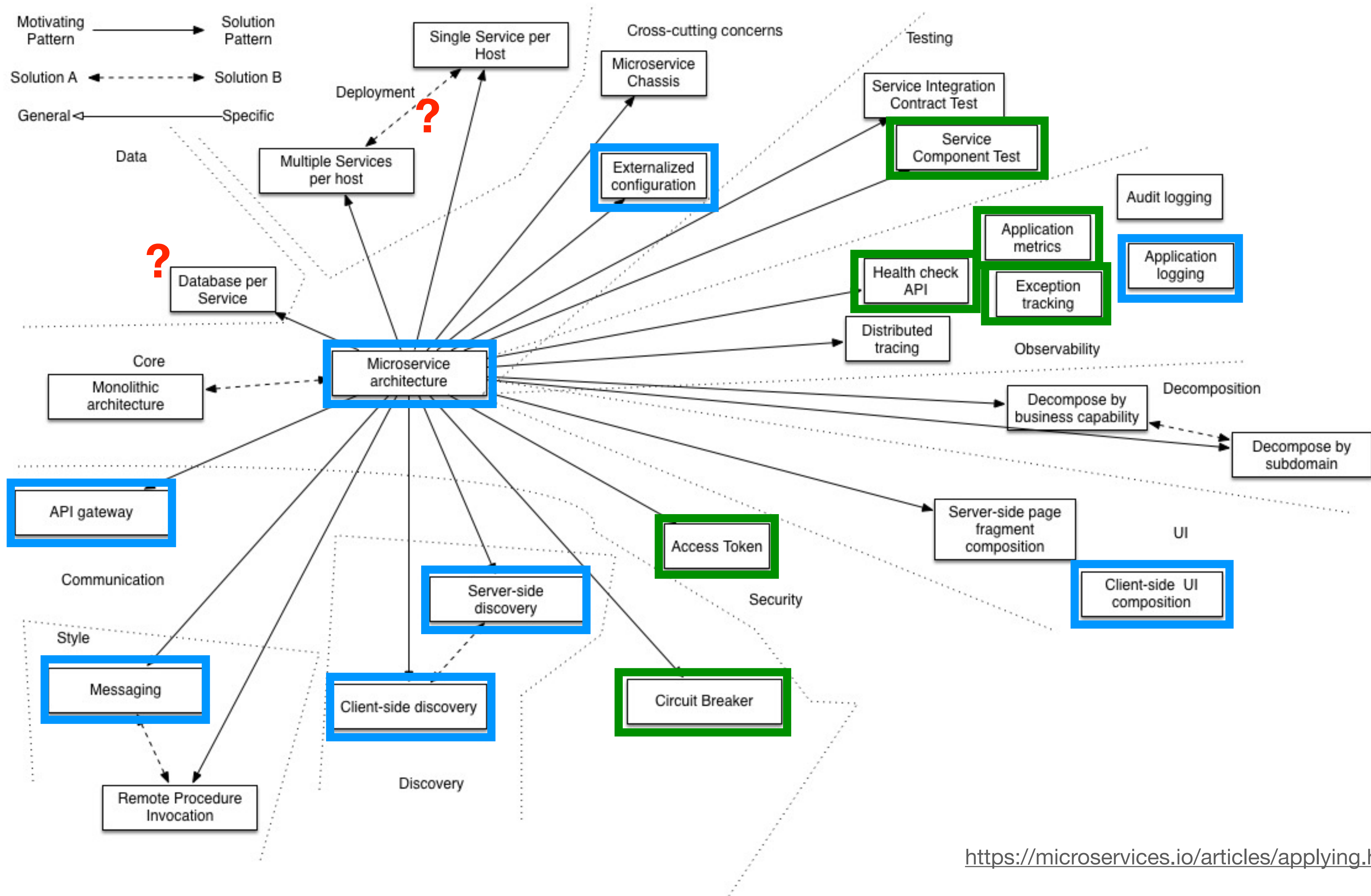
```
(def app-db
  {:widgets {:line-chart-demo {:components [[:<chart-1> :<chart-2>]
                                             [[:<chart-3> :<chart-4>]]
                                             :blackboard [[:<chart-1> :selected] "pv"]}
    :<chart-1>  {:tab-panel {:value :<chart-1>/config
                           :data-path [[:<chart-1> :tab-panel]]
                           :grid {:include true :strokeDashArray {:d 3 :g 3}
                           :stroke "#ffffff"}
                           :x-axis {:include true :dataKey ""
                           :orientation :bottom :scale "auto"}
                           :y-axis {:include true :dataKey "" :orientation
                                   :bottom :scale "auto"}
                           :legend {:include true :layout :horizontal
                                   :align :center :verticalAlign :bottom}
                           :tooltip {:include true}
                           :selected "pv"
                           :pub [[:<chart-1> :selected]]
                           :sub []
                           :components []}
    :<chart-2>  {...
      :sub [[:<chart-1> :selected]]
      :components []}}})
```



Option 2
Each component exists
“inside” its container

```
(def app-db
  {:widgets {:line-chart {:components [[:<chart-1> :<chart-2>]]
    :<chart-1> {:tab-panel {:value :<chart-1>/config
      :data-path [:<chart-1> :tab-panel]}
      :grid {:include true :strokeDashArray {:d 3 :g 3}
        :stroke "#ffffff"}
      :x-axis {:include true :dataKey ""
        :orientation :bottom :scale "auto"}
      :y-axis {:include true :dataKey "" :orientation
        :bottom :scale "auto"}
      :legend {:include true :layout :horizontal
        :align :center :verticalAlign :bottom}
      :tooltip {:include true}}}
    :<chart-2> {:tab-panel {:value :<chart-2>/config
      :data-path [:<chart-2> :tab-panel]}
      :grid {:include true :strokeDashArray {:d 3 :g 3}
        :stroke "#ffffff"}
      :x-axis {:include true :dataKey ""
        :orientation :bottom :scale "auto"}
      :y-axis {:include true :dataKey "" :orientation
        :bottom :scale "auto"}
      :legend {:include true :layout :horizontal
        :align :center :verticalAlign :bottom}
      :tooltip {:include true}}}}})
```

container
component
context-data



Coverage Plan

Targets

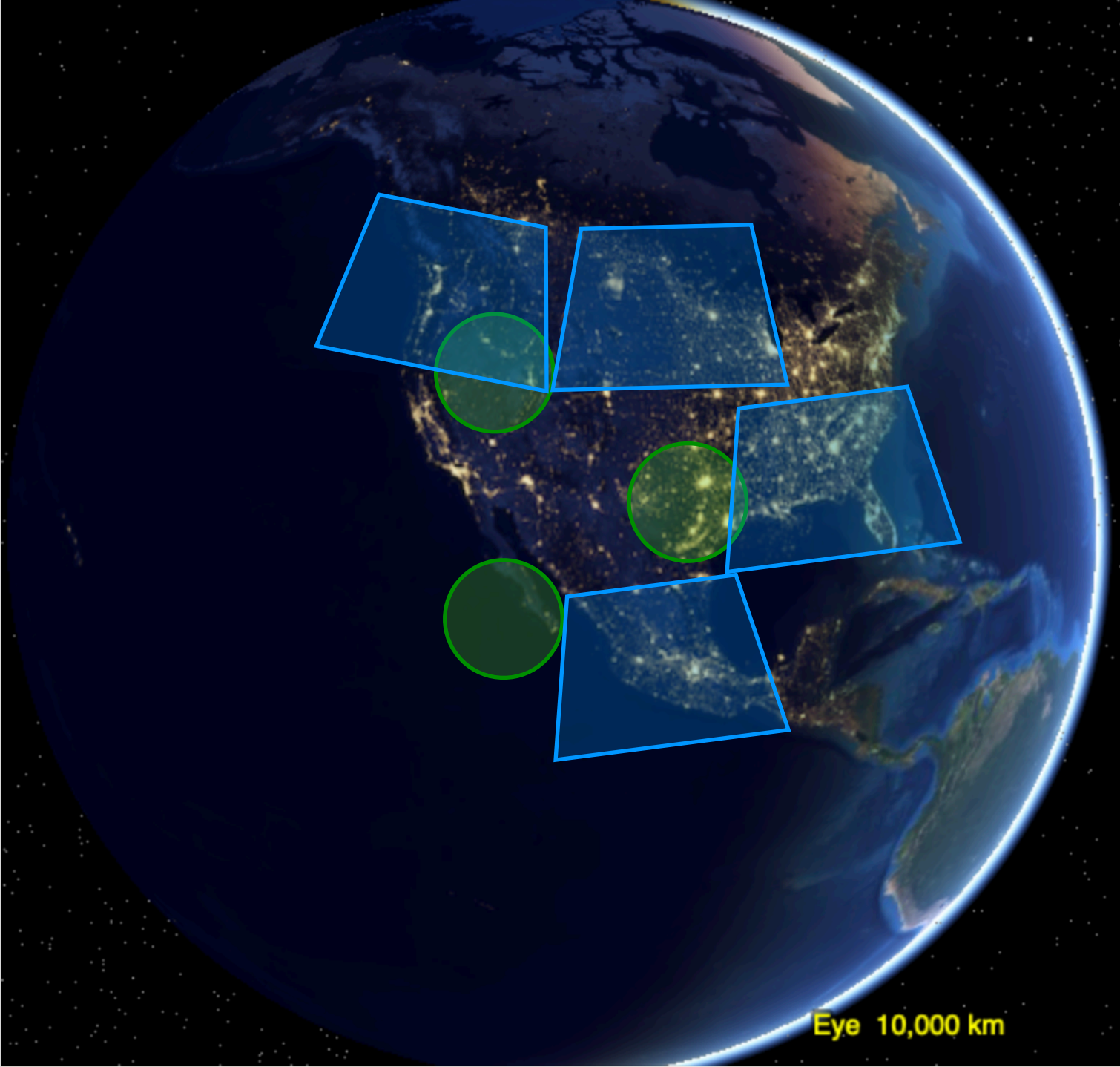
Code	Region	Name	Email
0	South	Frea	frea@widget.org
1	North	Barry	barry@conversation.com
2	West	Mousa	mousa@conversation.com
3	South	Mabel	mabel@example.com
4	East	Khabat	khabat@deck.com

Satellites

Code	Region	Name	Email
0	South	Frea	frea@widget.org
1	North	Barry	barry@conversation.com
2	West	Mousa	mousa@conversation.com
3	South	Mabel	mabel@example.com
4	East	Khabat	khabat@deck.com

Current Time

Globe



Sat Jan 15 2022 07:00:00 GMT-0500 (EST)

remote
data-source

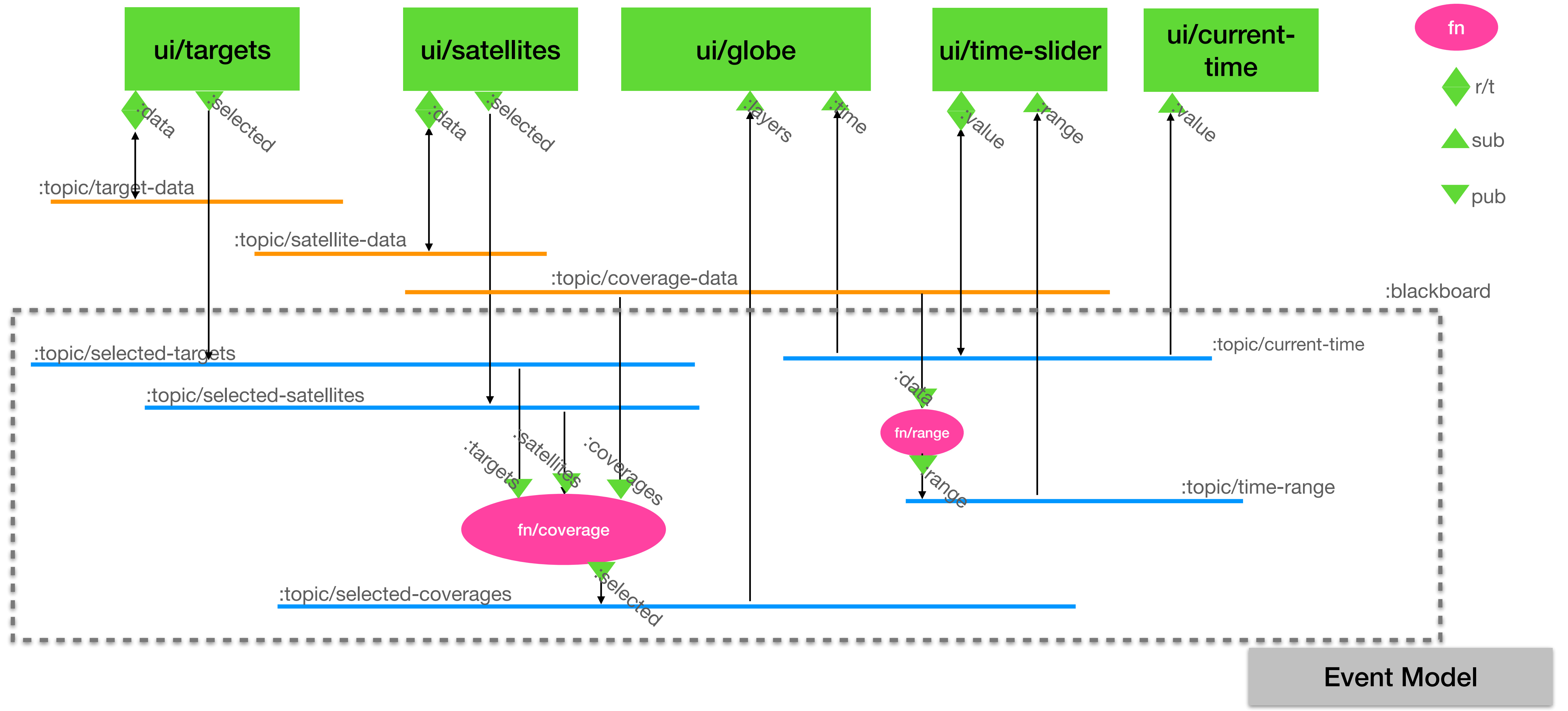
local
blackboard

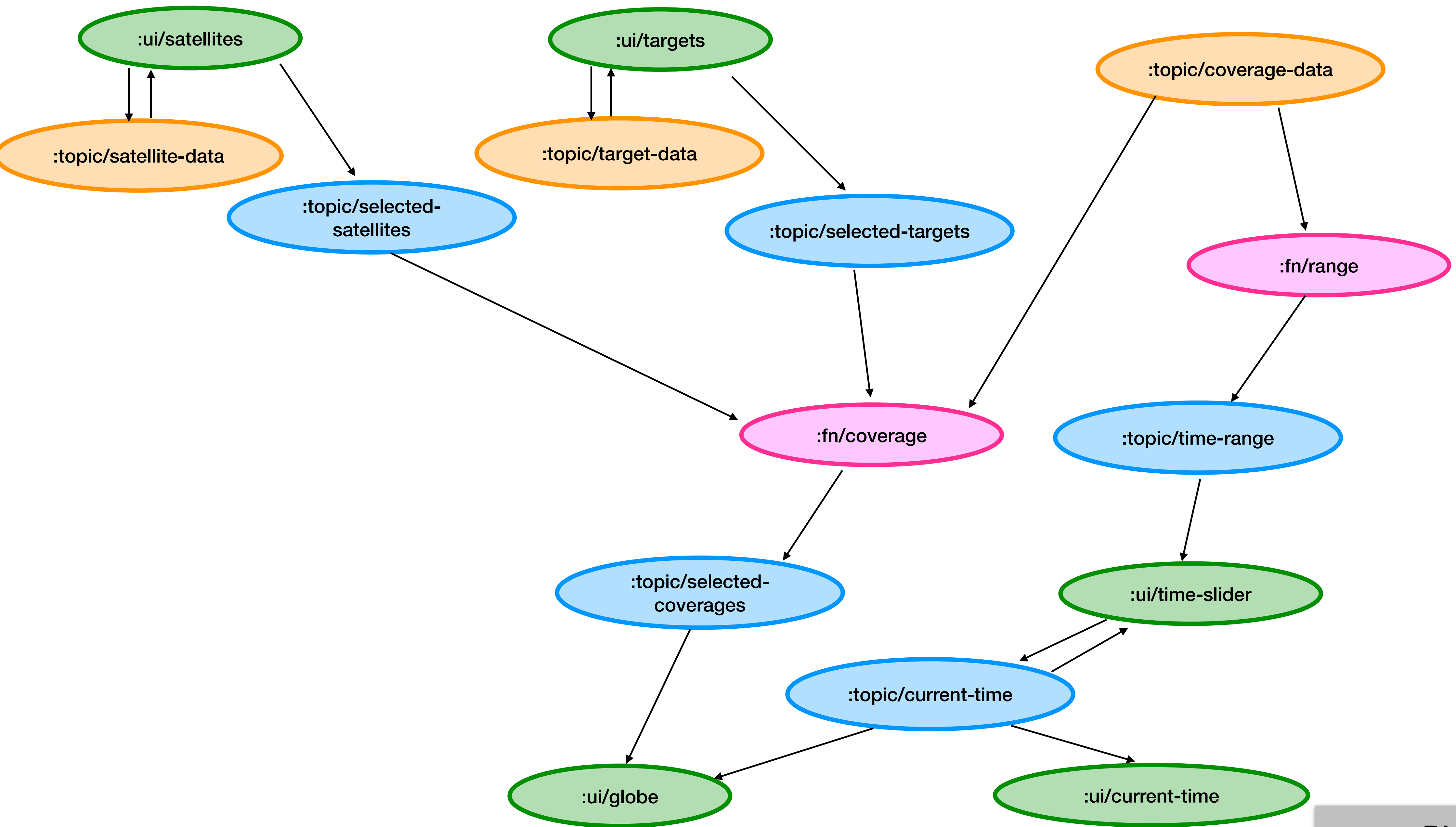
fn

r/t

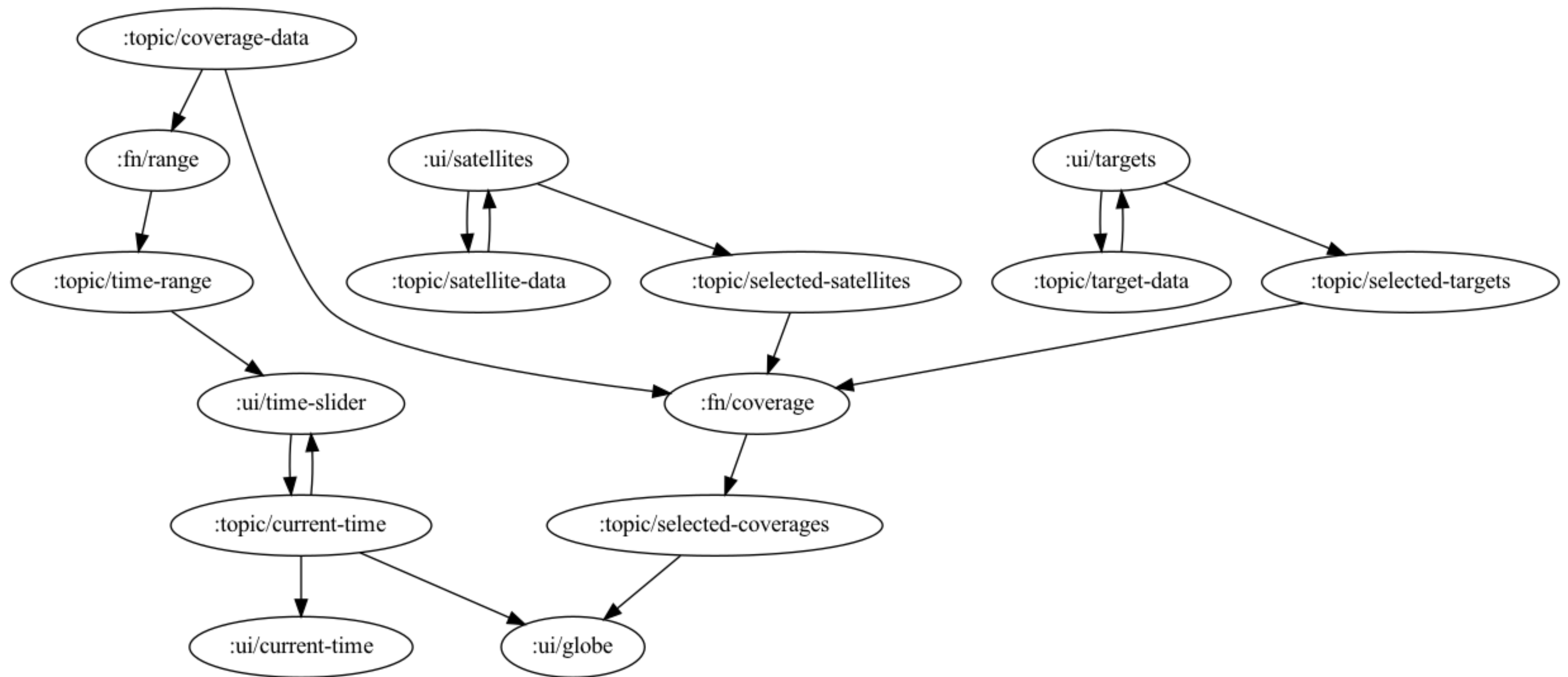
sub

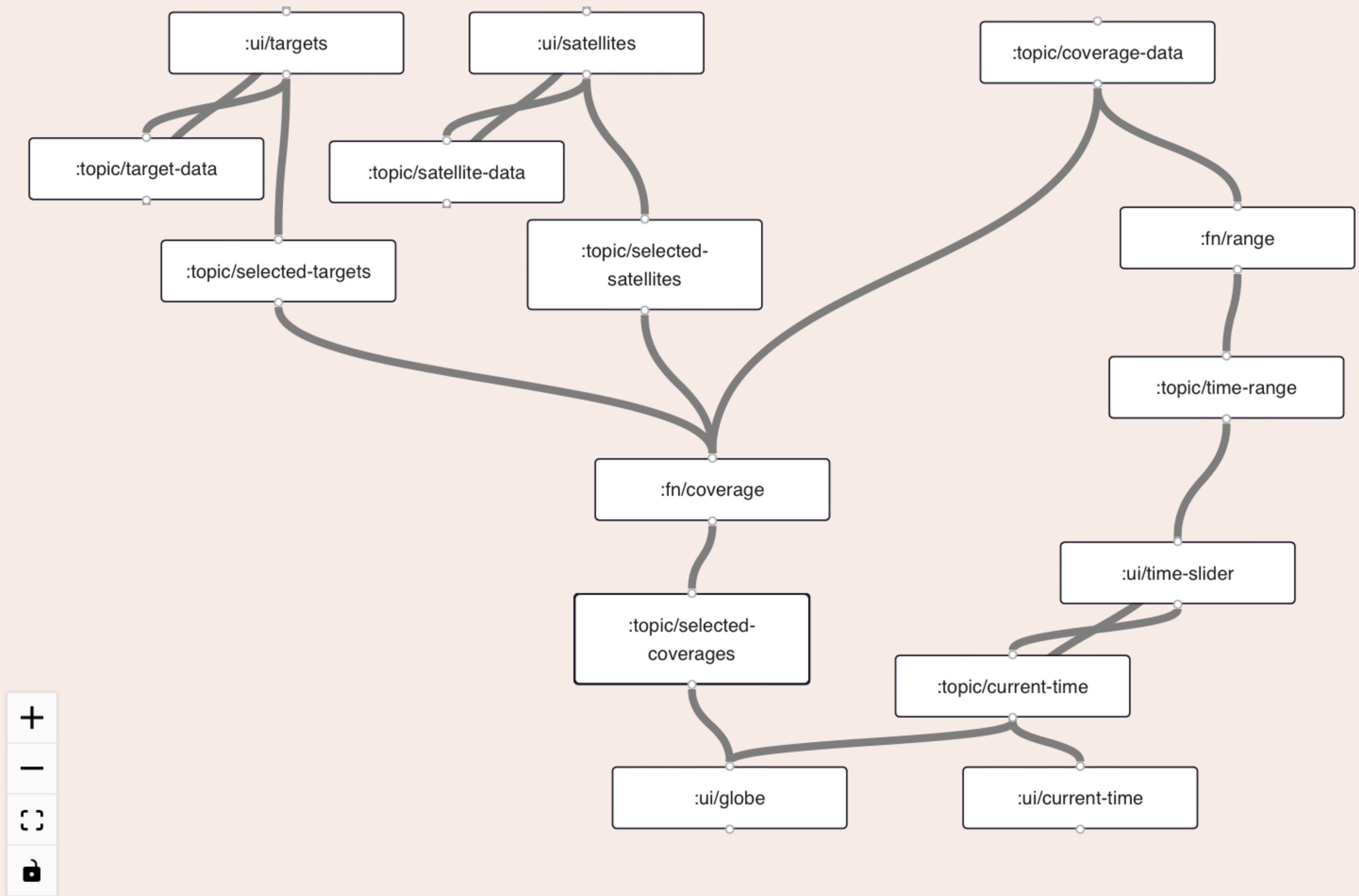
pub

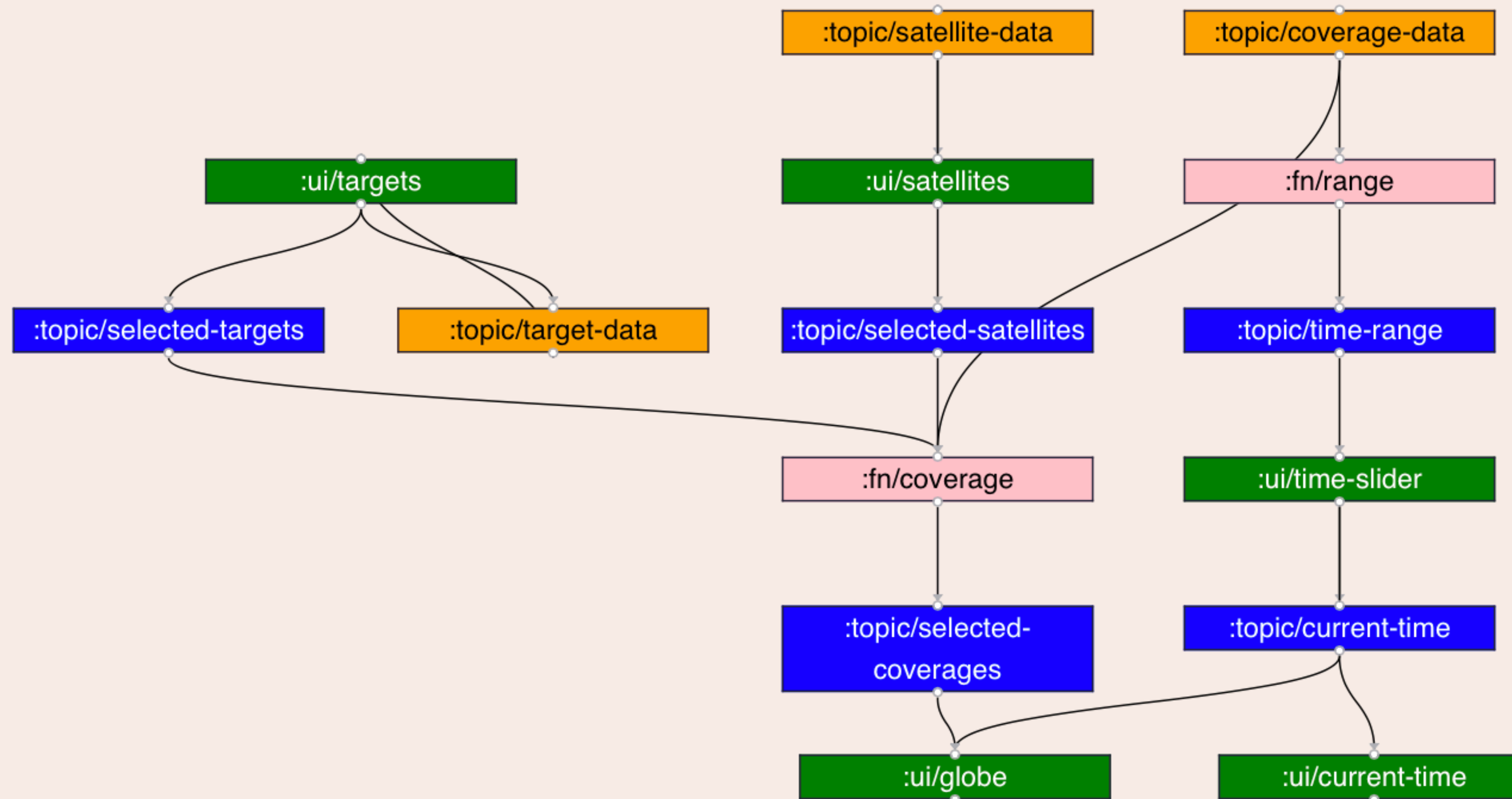




Digraph







flow w/colors