Controlling flows execution

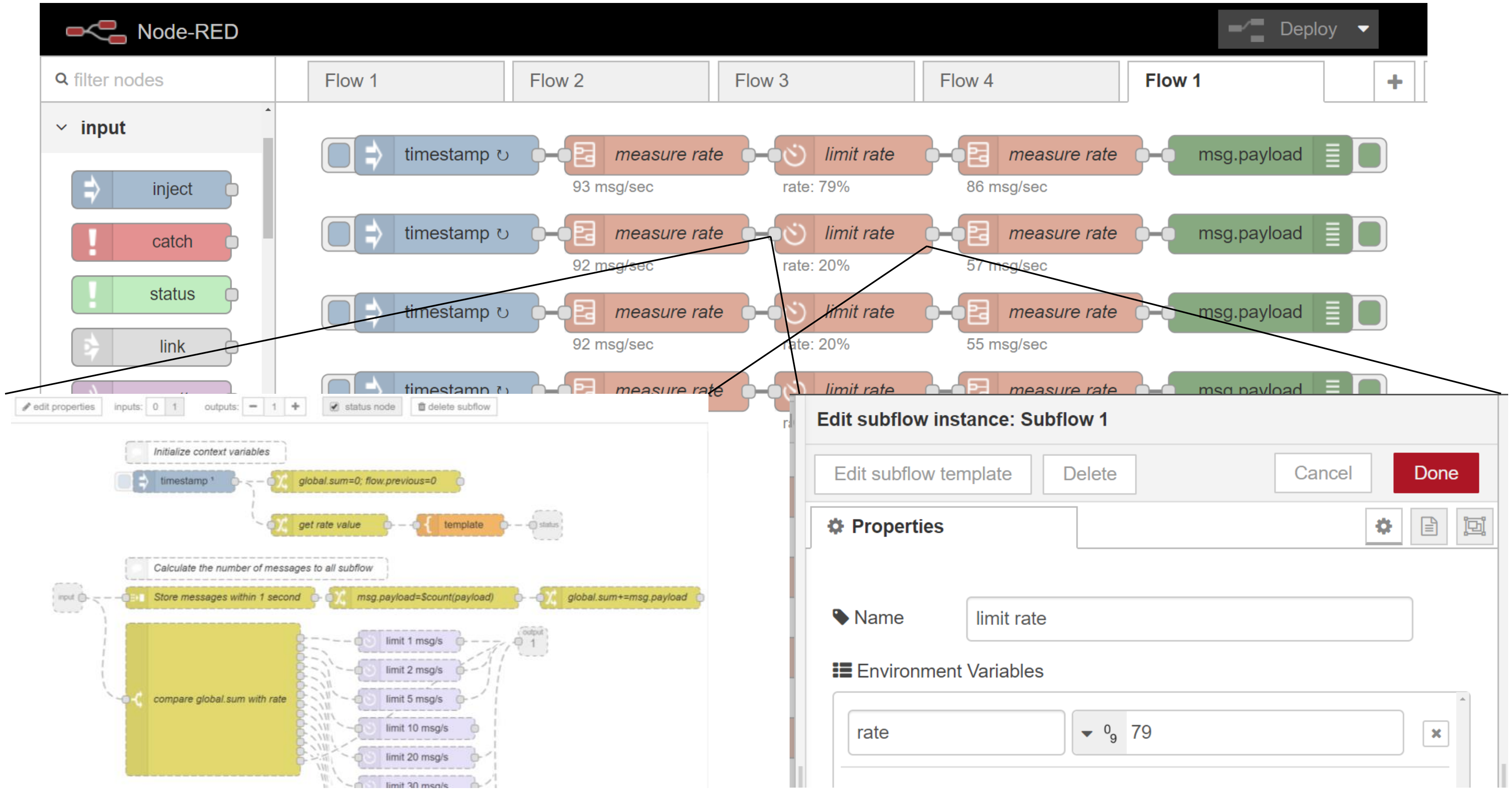
- We'd like to control the number of messages or priority in flows. For example, a flow use 50% CPU time and other flows share another 50% CPU time when flows handle a lot of messages.
- There will be a risk that Node-RED process is stopped when a user simply set low priority to a flow which retains a lot of messages in the queue.
- As another solution, incoming messages may have to be limited based on the size of queue in each node of the flow.

#	Methods	Challenge level	Priority setting using existing function	Risk of queue overflow
1	EventEmitter extension to support priority	Difficult (We need to discuss it with Node.js community)	None (implementation is needed)	High
2	Flow control using delay node	Easy	Available (Hitachi already implemented it)	High
3	Multi runtime	Middle (We need to define specifications)	Available (nice command in Linux)	Middle (It mitigates the risk but it cannot avoid the risk totally)
4	Incoming message control	Middle (We need to define specifications)	None	None

There're details in the next slides

Flow control using delay node

- We developed subflow node using delay nodes to set the relative limit between flows.
- While testing the subflow, we found that the Node-RED process is crushed when the message queue is full.



Name: limit rate

Environment Variables

rate | 79

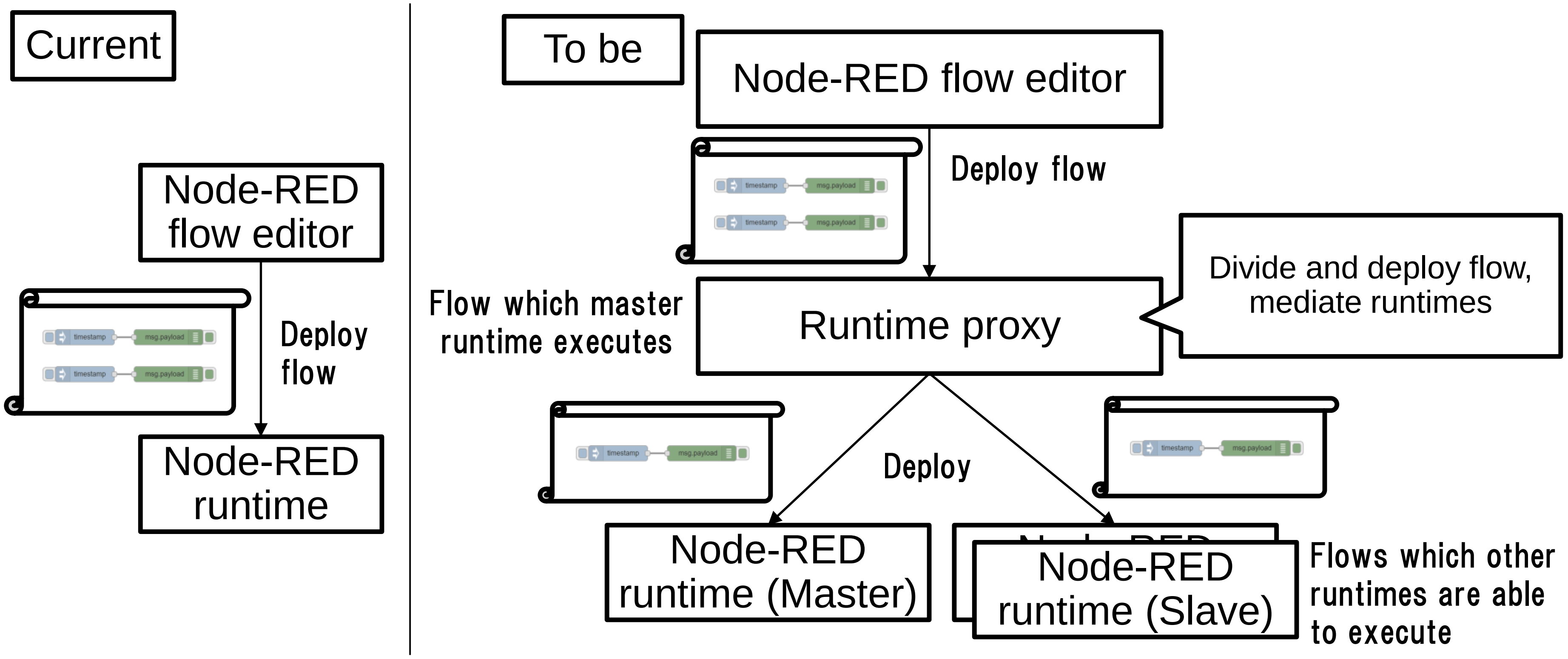
Flow inside subflow

Subflow property to set relative rate

Multiple runtime

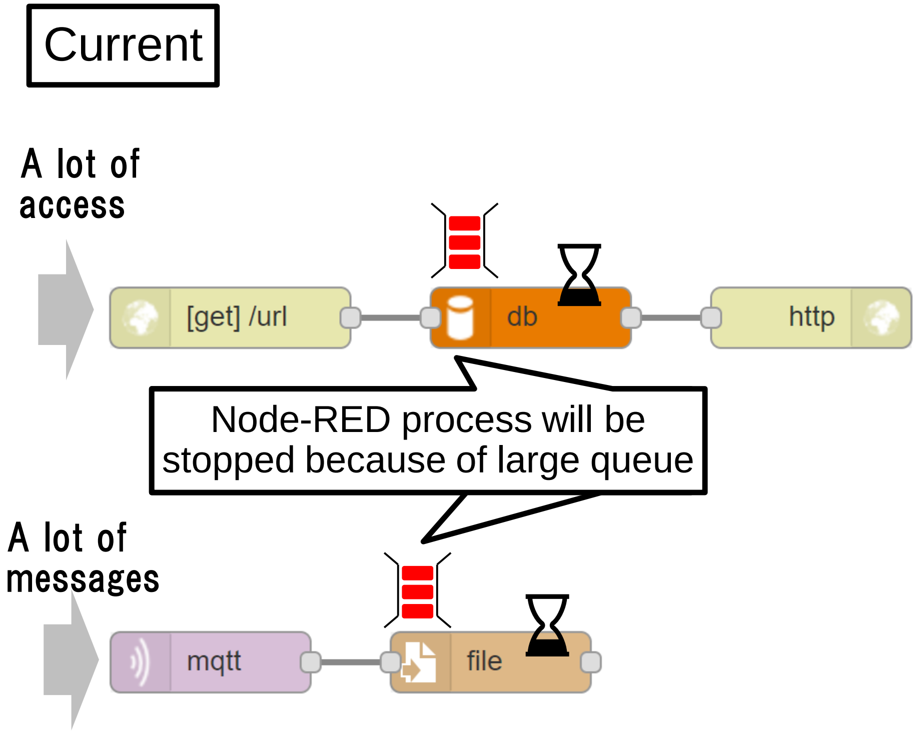
HITACHI
Inspire the Next

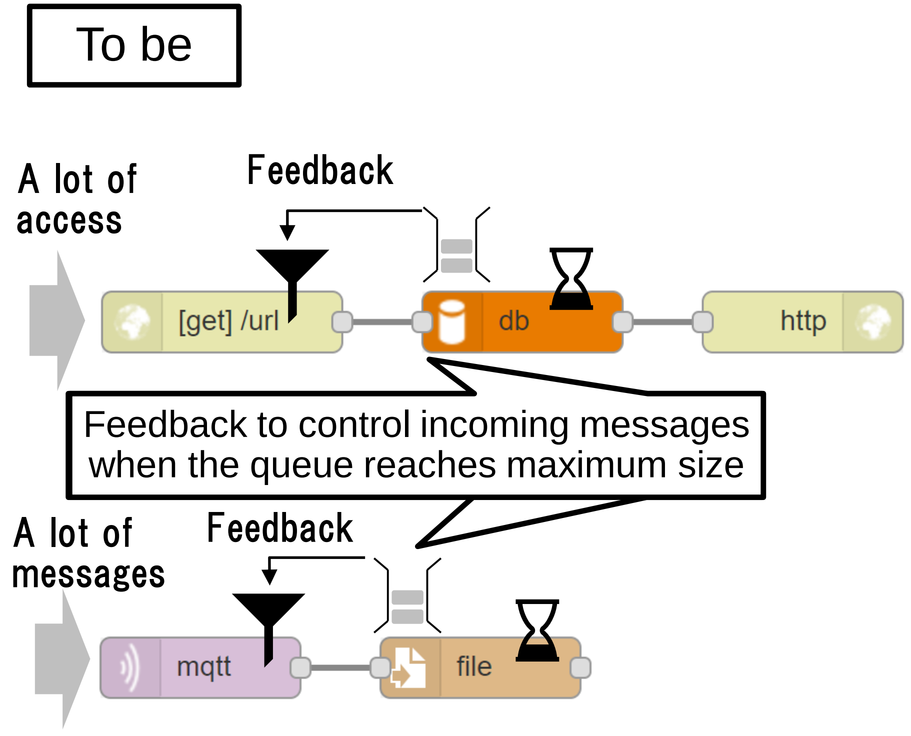
- To control flow priority using nice command and utilize CPU resources efficiently, we'd like to use multiple runtimes to execute each flow in different processes.
- To manage multiple runtimes, runtime proxy divides flow data for each multi runtime from one flow JSON data and it mediates runtimes.
- This method mitigates the risk of queue overflow but it cannot avoid the risk totally.



Incoming message control

- To avoid the risk of message queue overflow, the incoming messages may have to be limited at the first node in the flow.
- We'd like to suggest feedback functionality in Node-RED to limit the incoming message based on the size of queue in the following nodes.


Current
A lot of access
[get] /url
db
http
Node-RED process will be stopped because of large queue
A lot of messages
mqtt
file


To be
A lot of access
Feedback
[get] /url
db
http
Feedback to control incoming messages when the queue reaches maximum size
A lot of messages
Feedback
mqtt
file