HITACHI
Inspire the Next

Node installation from other than public site

Kazuhiro Ito

Controlling flows execution

HITACHI
Inspire the Next

- 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.

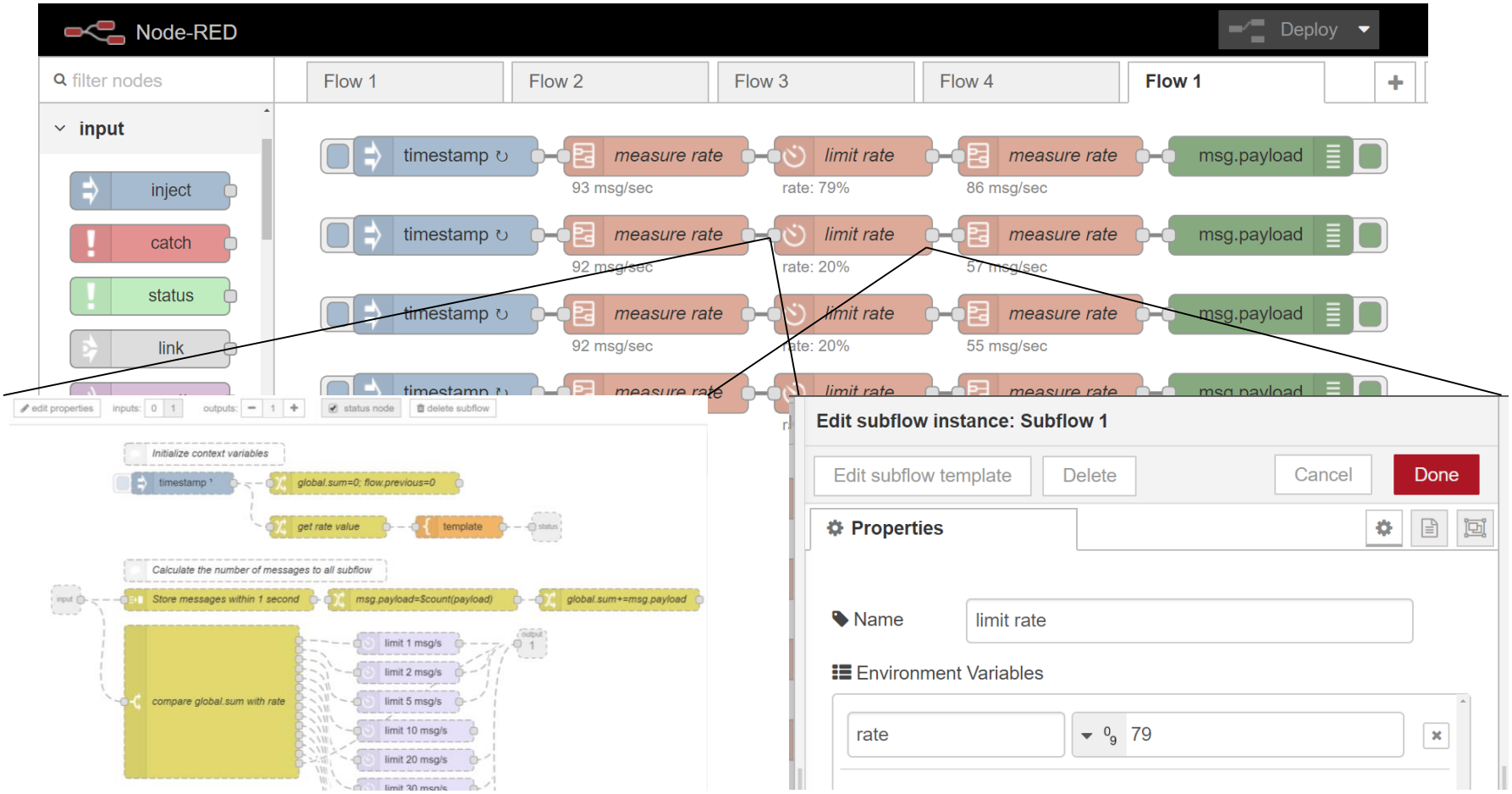
#	Methods	Challenge level	Priority setting using existing function	Risk of queue overflow
1	EventEmitter extension to support priority	Difficult (We need to discuss it with Node.js community)	None (implementation is needed)	High
2	Flow control using delay node	Easy	Available (Hitachi already implemented 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

HITACHI
Inspire the Next

- 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.

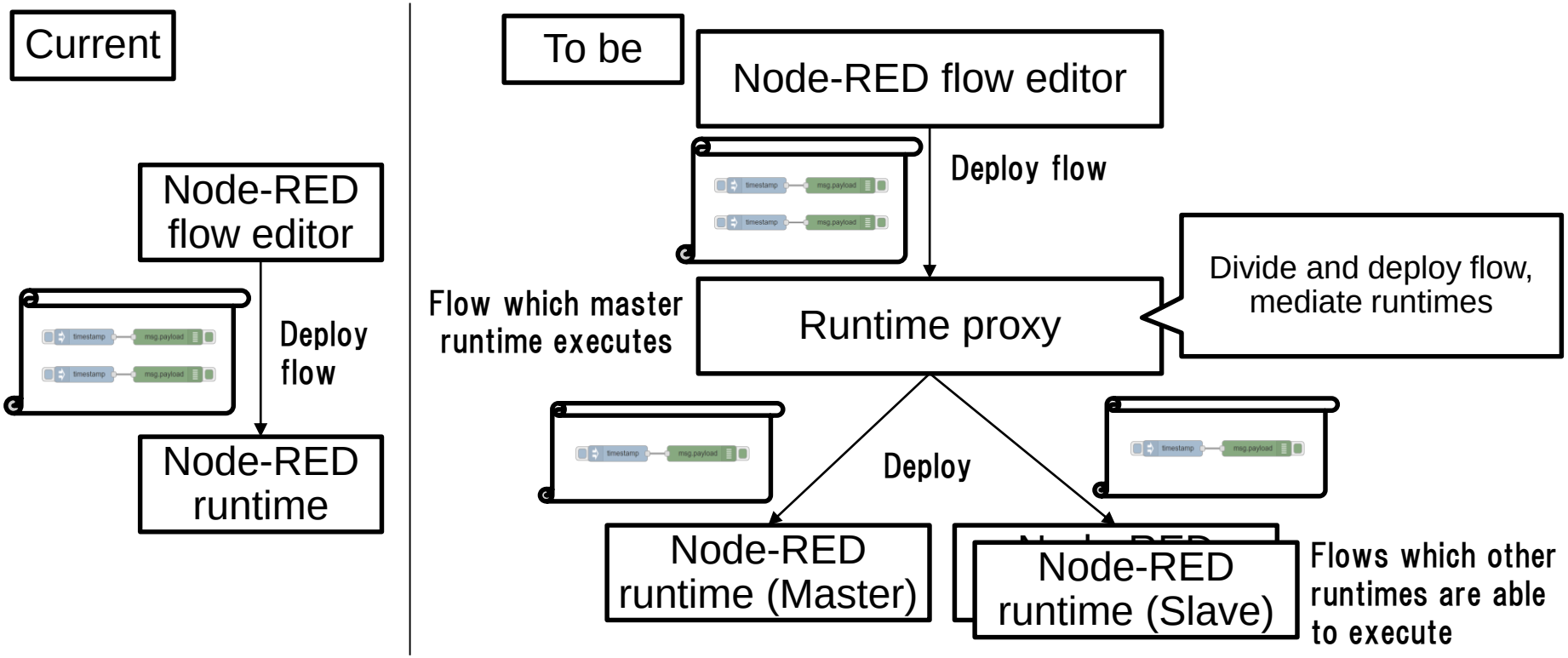


Flow inside subflow

Subflow property to set relative rate

Multiple runtime

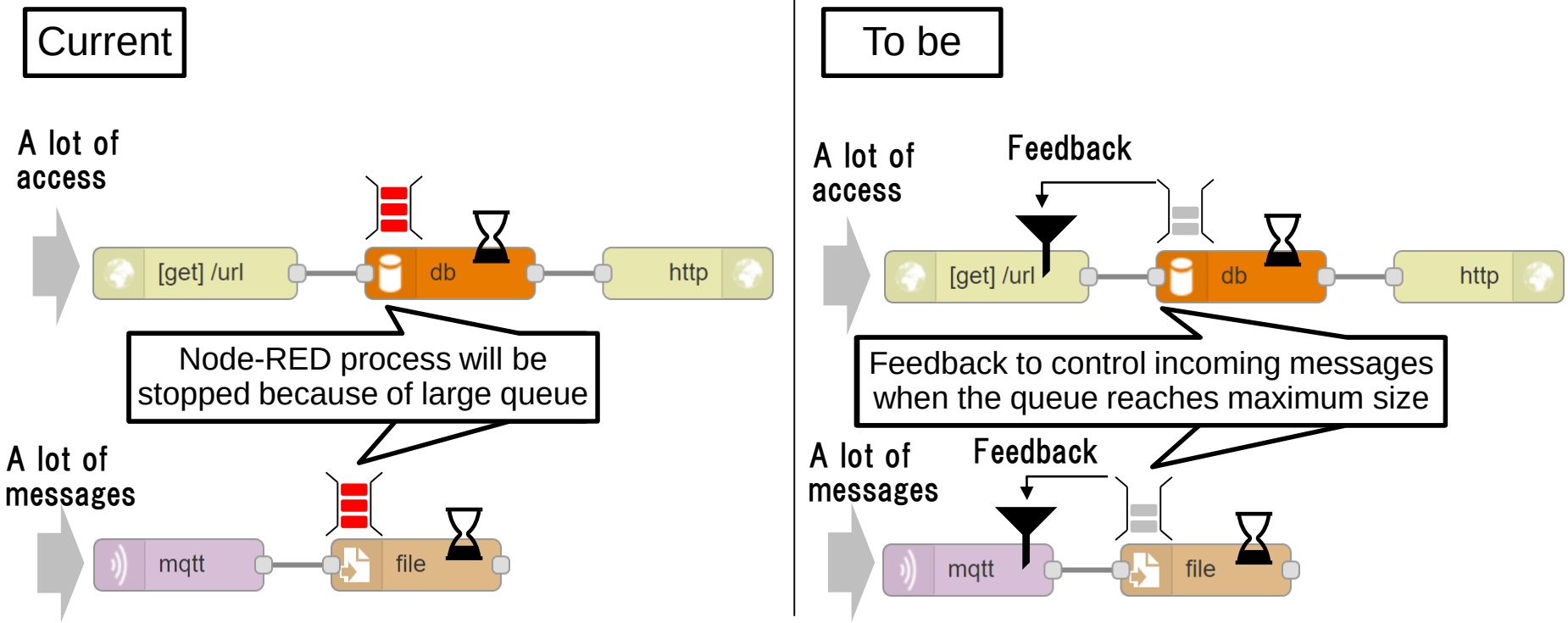
- To control flow priority using nice command and utilize CPU resources efficiently, we'd like to use multiple runtimes to execute each flow in different processes.
- To manage multiple runtimes, runtime proxy divides flow data for each multi runtime from one flow JSON data and it mediates runtimes.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



Incoming message control

HITACHI
Inspire the Next

- 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.



Node installation from other than public site

HITACHI
Inspire the Next

■ Summary

Using the function for adding nodes of "Palette", you can install your own node on Node-RED from other than public site, such as local environment.

You can install your own node from other than the public site by Node-RED settings and preparing your own 'npm' registry. This time we are considering providing more convenient and easier install function. We propose ideas below.

■ Ideas

#	Idea	explanation	Delivery Site	modification
1	Specifying URLs for node information catalogues	Allowing to add Git and tar file URLs in the catalogue.json	Need GitHab(GitLab) or Web Server	small
2	Specifying URL for node information on "Palette"	Allowing to input the URLs referring a remote Git repository and a tar file on "Palette", and installing a node.	Need GitHab(GitLab)orWeb Server	medium
3	Specifying the path of local tar file on "Palette"	Allowing to input the path of local tar file, and installing a node.	Not Need Obtaining files	large

Specifying URLs for node information catalogues

HITACHI
Inspire the Next

■ Idea 1

Users can add the URLs referring a remote Git repository or a tar file directly in the catalogue.json

catalogue.json:

```
{
    "description": "Additional node example.",
    "keywords": [
        "node-red",
        "example"
    ],
    "types": [
        "example"
    ],
    "official": false,
    "updated_at": "2019-06-21T13:11:36.035Z
    "id": "node-red-contrib-example",
    "version": "0.1.0",
    "git-url": "https://testsrv/node-red-contrib-example.git"
},
    :
```

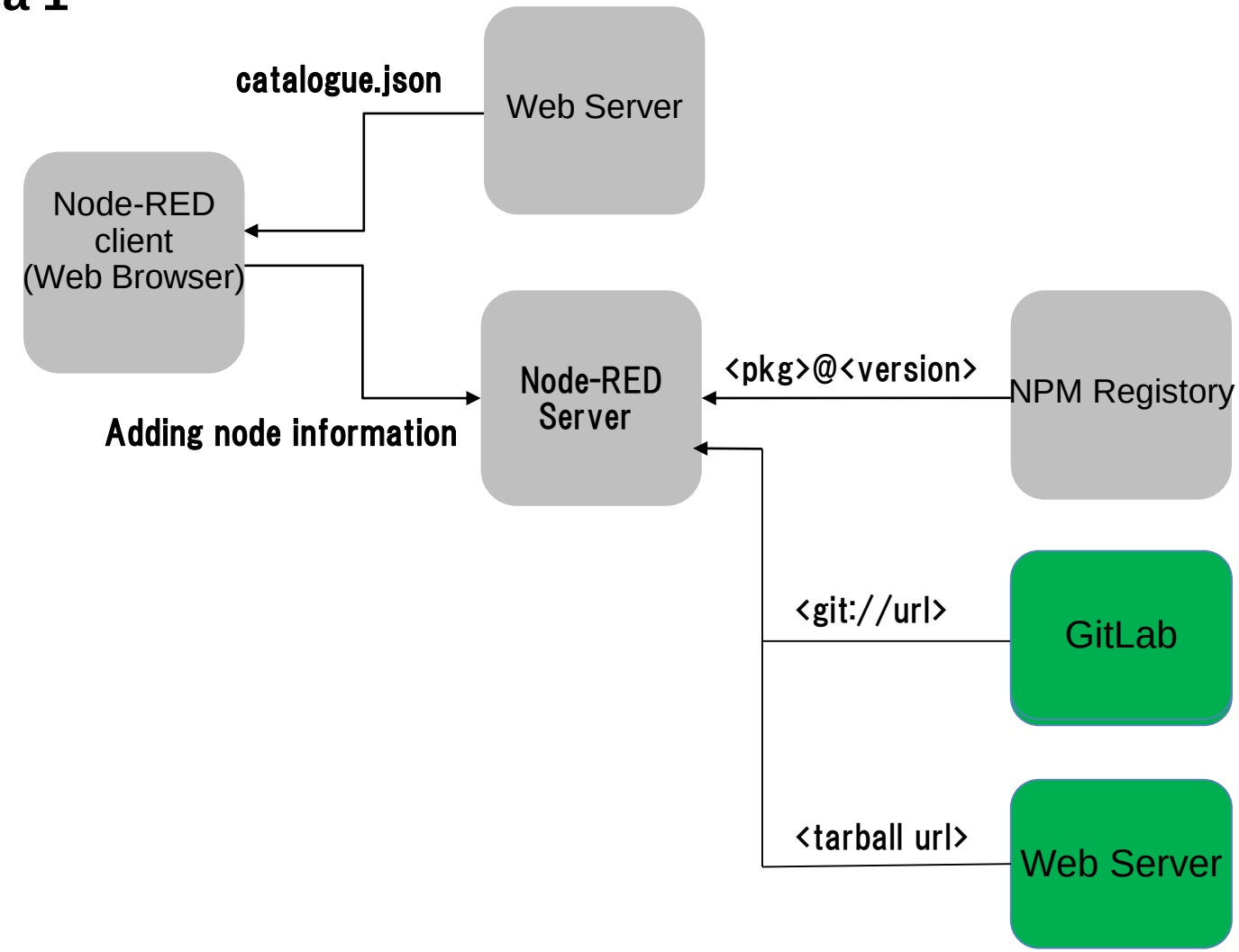
Specifying the URLs referring a remote Git repository and a tar file in the catalogue.json.

npm install <git:// url> and
npm install <tarball url> are also OK

Specifying URLs for node information catalogues

HITACHI
Inspire the Next

■ Idea 1

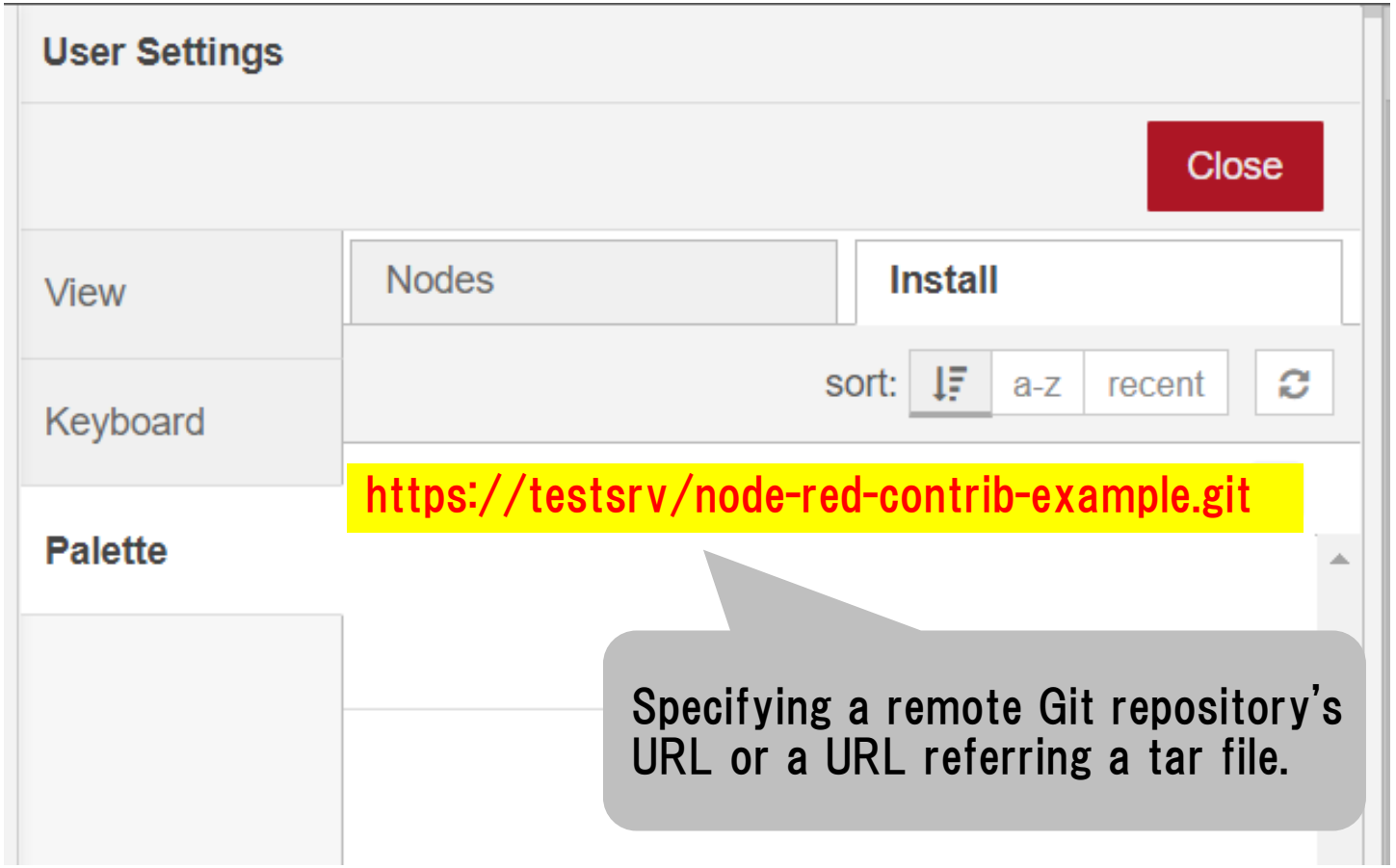


Specifying URL for node information on "Palette"

HITACHI
Inspire the Next

■ Idea 2

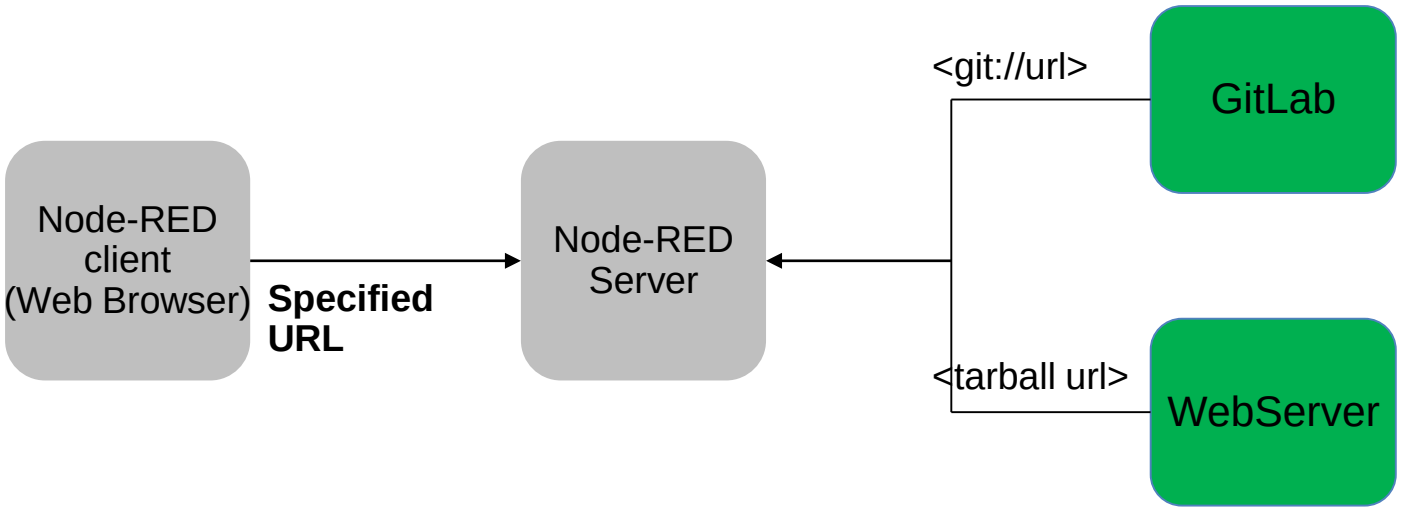
Users can input the URLs referring a remote Git repository and a tar file on "Palette"



Specifying URL for node information on "Palette"

HITACHI
Inspire the Next

■ Idea 2

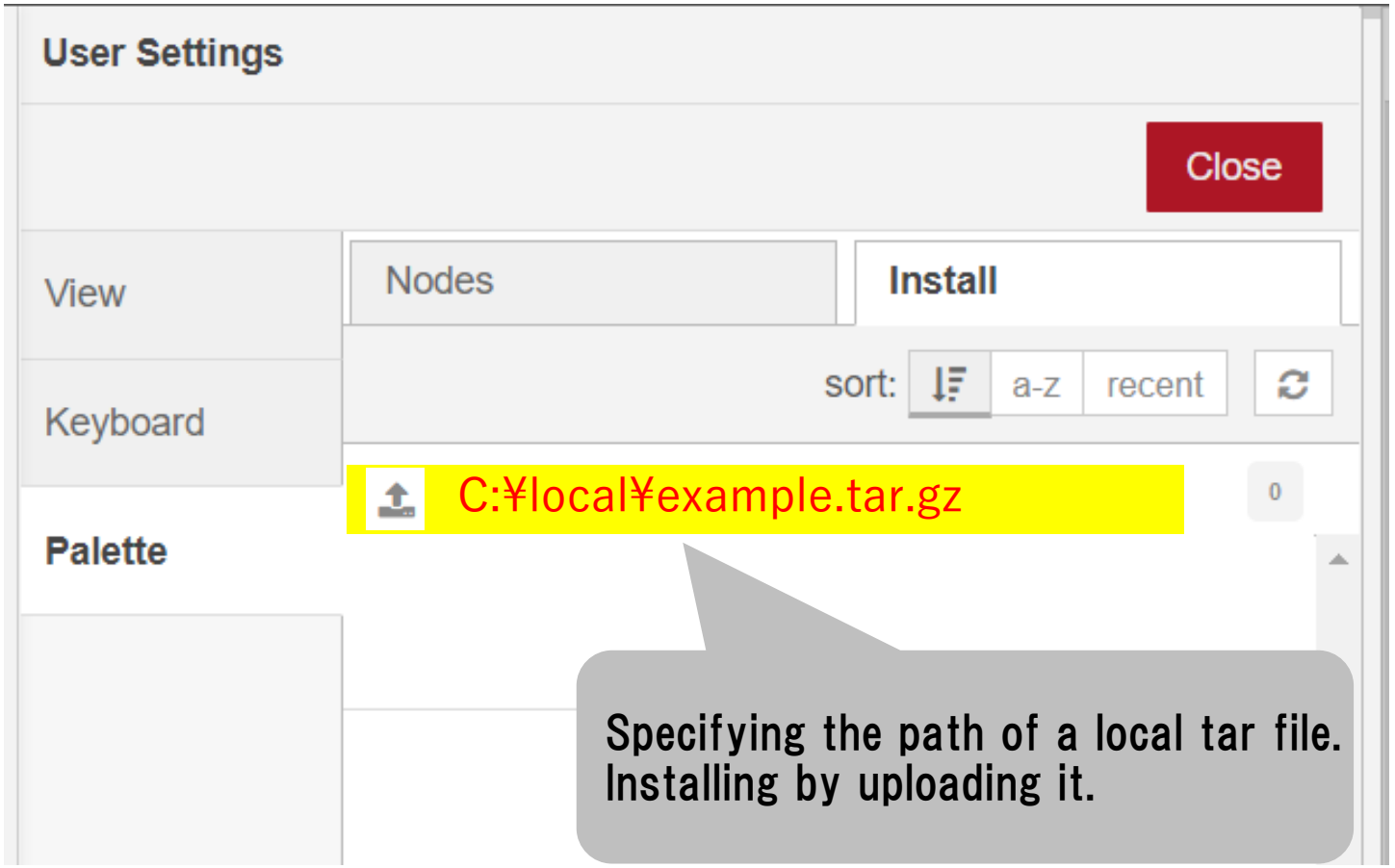


Specifying the path of a local tar file on "Palette"

HITACHI
Inspire the Next

■ Idea 3

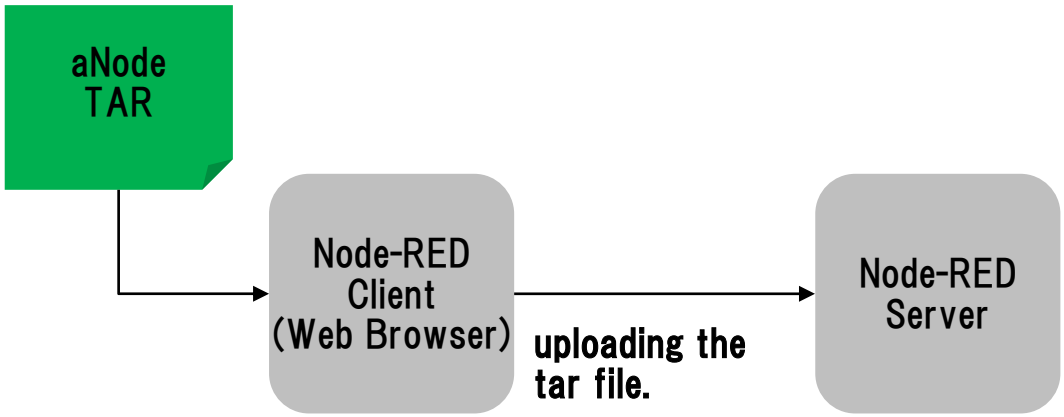
Users can input the path of a local tar file.



Specifying the path of a local tar file on "Palette"

HITACHI
Inspire the Next

■ Idea 3



HITACHI

Inspire the Next