
Node-RED

Who Is It for?

Within organisations there are a number of users who are moving beyond consuming business services and into creating them.

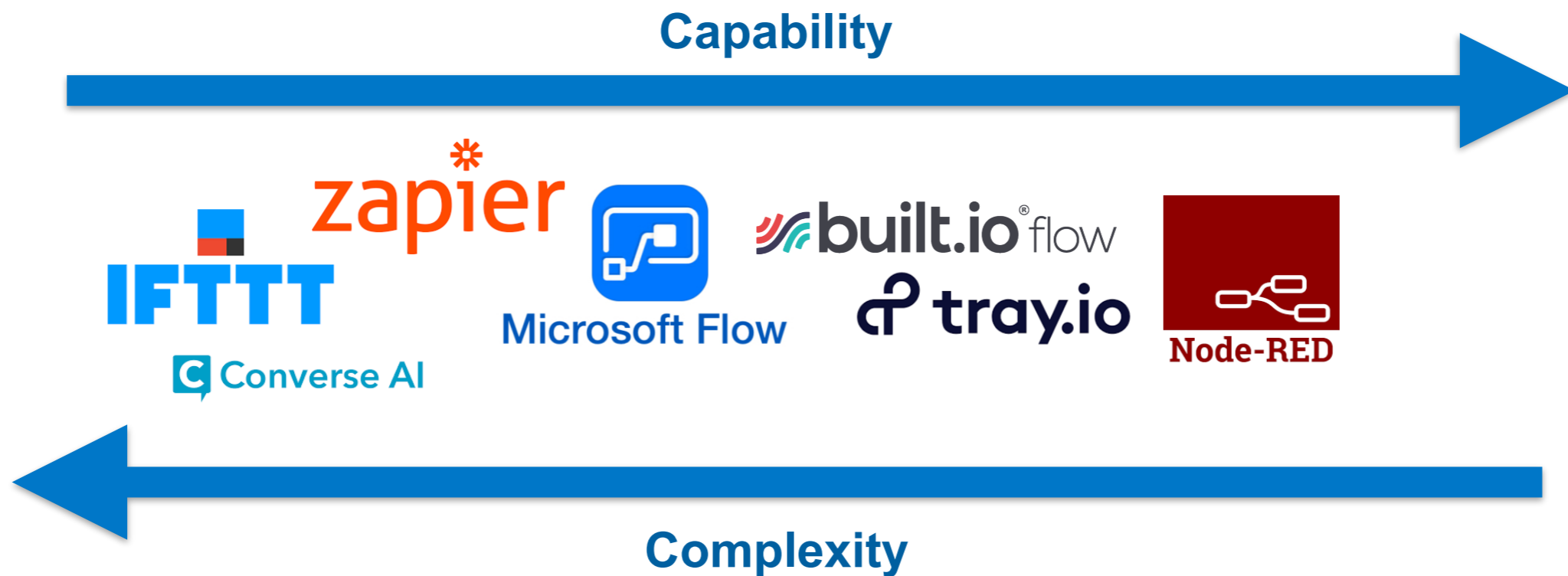
Complex business processes have been built using tools like Excel as it was the only option available.

The world of business applications has moved on, cloud platforms offer the ability for users to customise their tool and workflows

Companies are offering platforms to tie services together without needing the skills of a developer

We call these people Builders

The Space



Node-RED

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things

Node-RED has a library of around 2000 packages for various services, such as Salesforce, IBM Watson, Zendesk, Slack and more.

Node-RED offers a powerful, fast tool for prototyping and experimentation with Programmable Communications, It's clear visual representation breaks down the barrier of code for builders and non-developers.

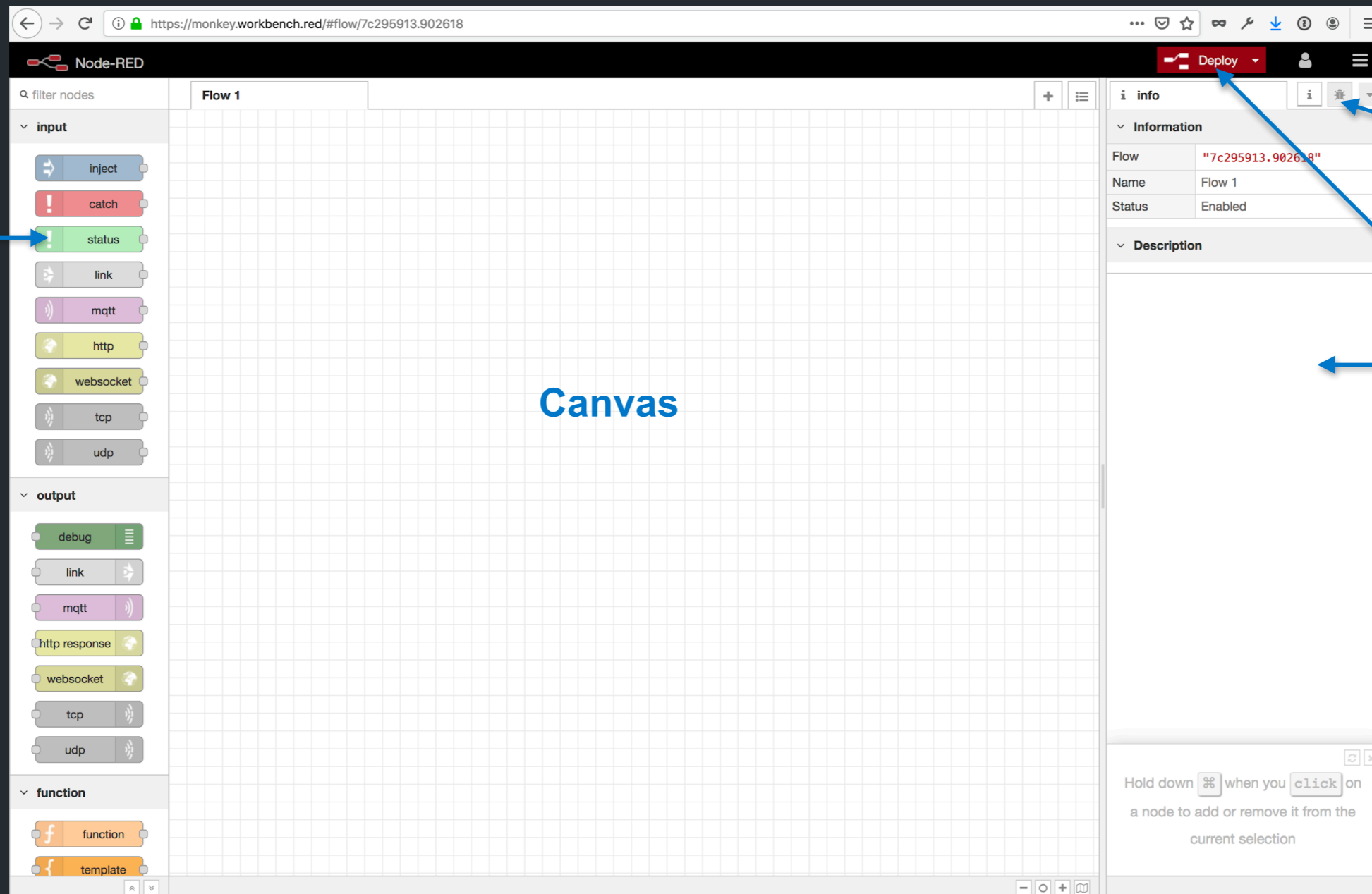
How Does It Work

At its core Node-RED is a web application comprised of an editor and a runtime, the platform is built in Javascript.

The editor runs in a regular web-browser, the user is presented with a canvas on which to arrange nodes from the palette on the left hand side.

By default Node-RED includes a standard palette of nodes, additional ones can be added from the menus.

The Editor



Menu

Debug

Deploy

Info

Palette

Canvas

Nodes

A node represents a function, we have (mostly) mapped our API calls to nodes.

Nodes are either, input, processing or output nodes.



A processing node can have multiple output paths

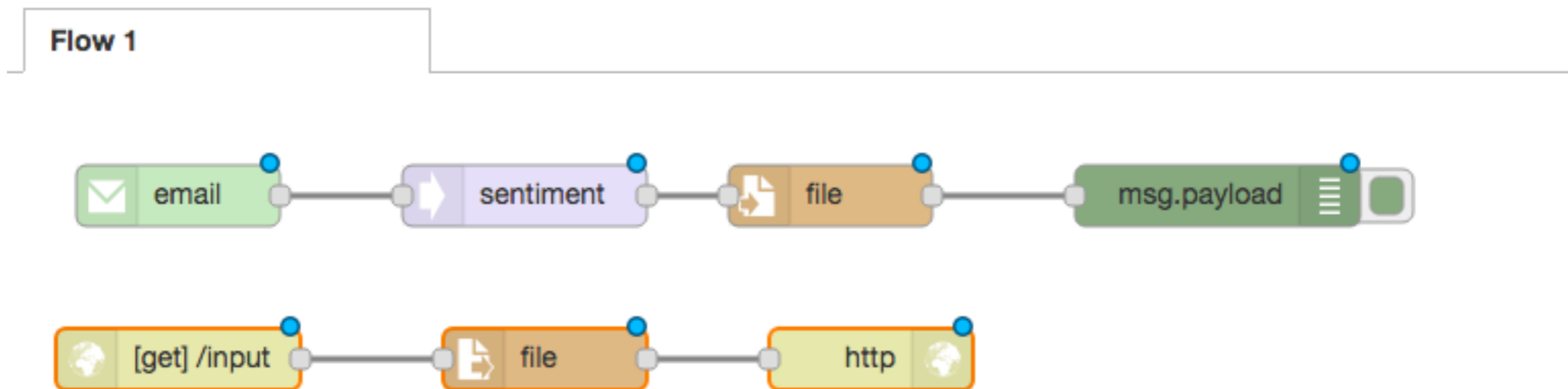


Some input nodes require a corresponding output node, for example the HTTP node



Flows

A flow is a slightly muddled term within Node-RED, it can either mean a string of nodes connected together into a path or it can refer to a tab within the canvas, the flow namespace is unique to the tab.



Messages

A message is an object passed between nodes in a flow, the `msg` object is created by an input node and is then forwarded, or modified by processing nodes.

There are certain conventions that allow nodes to talk to each other, the most common way that data is passed from one node to another is in the `msg.payload` parameter

The Runtime

The runtime is the backend Node.js process which executes flows that have been deployed from the editor.

The editor does not need to be open for the runtime to process a flow.

The runtime is a long lived process and as such is always available

The runtime can expose external servers eg HTTP, TCP, MQTT etc or invoke flows based on its own clock and scheduler.

Questions?

<https://nodered-workshop.nexmodev.com>