

Infrastructure as Code and No-Code/Low-Code Platforms in an IIoT context

by Burkhard Heisen

Understanding and leveraging the possibilities of Industry 4.0 has become critical to defending hard-fought market positions and proving new leadership in fast evolving manufacturing markets. Currently, six out of ten industrial enterprises in Germany already apply industry 4.0 practices and this tendency is steadily increasing (Bitkom e.V., 2020). For instance, eighty percent of German SMEs are planning or budgeting upcoming IoT projects (PAC Deutschland, 2019). But to enable agile and powerful IIoT-driven business, IT-departments must choose tools that support them as best possible. Most solutions in the market are based on two very different approaches: A No-Code / Low-Code Platform (NC/LC) or Infrastructure as Code (IaC). Each has their own characteristics and lead to very different operational realities for the people responsible for IIoT architectures. Here's what you need to know to choose what's best for you.

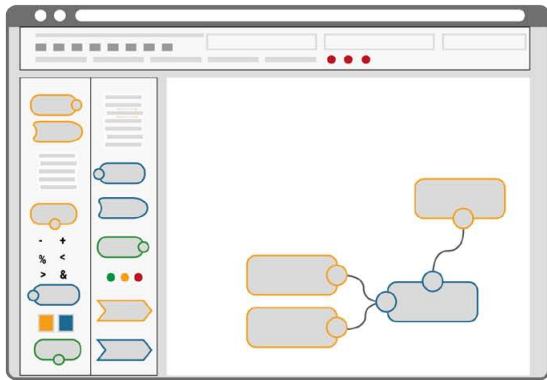
What it means

Before delivering a precise definition of IaC and NC/LC, it is important to take a step back and examine the basic term infrastructure in the IIoT context.

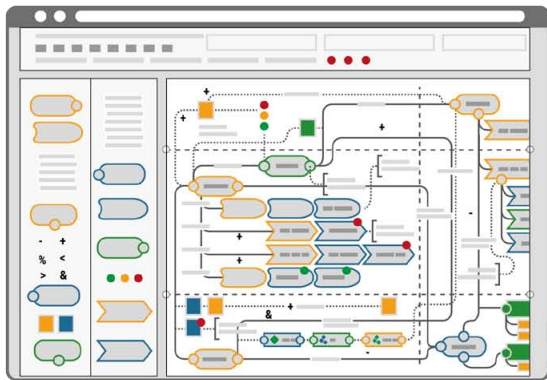
Let's look at a traditional manufacturing enterprise with two production machines A and B. Each of them has a controller, which is connected to the companies' network C and communicates via different protocols D and E. The input data is then used by the different applications F and G to follow specific purposes, e.g. predictive maintenance. This most basic infrastructure already has seven elements. But in reality examples can easily be far more complex and may need to support growing or changing dynamics.

In a NC/LC based IIoT Edge Hub, all components A-G are visually represented and interrelated per drag-and-drop system on a solution design canvas. Each element is typically described via entry fields, check boxes and radio buttons in dedicated configuration windows. Immersing yourself in the code or into seemingly complex configuration files is not required. At the same time, this user convenience limits the deployment to those systems that fit into a graphical representation.

In contrast, the Infrastructure as Code (IaC) approach describes an entire infrastructure, needed for an IIoT use-case, in only one structured text file - often called the configuration or commissioning file. This file lists all of its components, called resources, and defines their specifications and interrelations in a standardized way. The commissioning file thus avoids time-consuming maneuvering of separate configurations or even scripts for each and all the different elements needed in a use-case. Everything is in one place, structured and standardized.

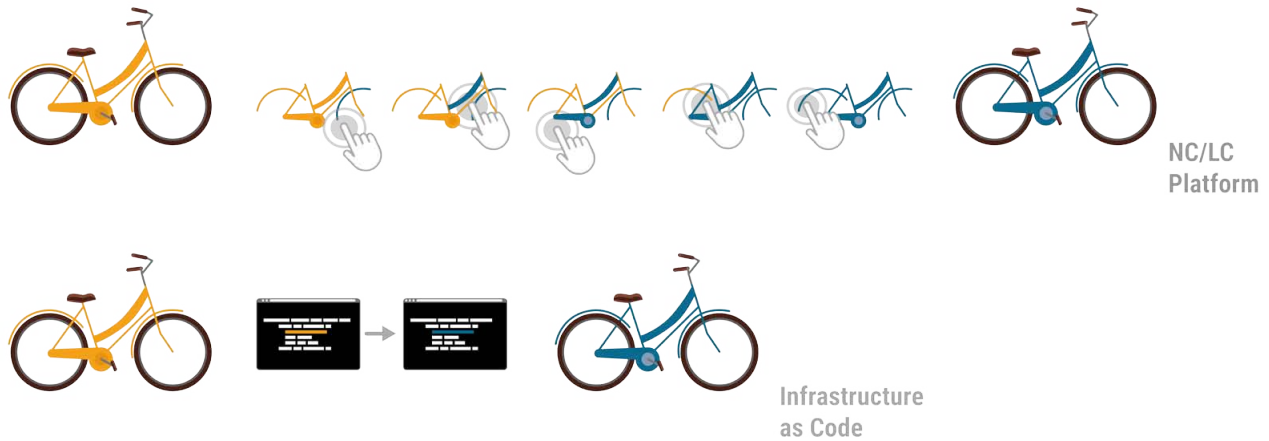


A simplistic No-Code / Low-Code Platform (NC/LC, left-hand side) in comparison to Infrastructure as Code (IaC, right-hand side).



A complex No-Code / Low-Code Platform (NC/LC, left-hand side) in comparison to Infrastructure as Code (IaC, right-hand side).

A key difference of the two approaches becomes clear by illustrating the workflow that's required when changes are needed. Imagine a bike that consists of a number of components, each defined by its name, size, colour, function, etc. Now imagine the colour of the bike needed to be blue instead of yellow. In the IaC approach the single text-based commissioning file can be searched for "colour" and the assigned codes can be auto-replaced in a single step. Alternatively "colour" could be a resource of its own and called upon by all components, thus allowing pre-defined auto-matching colour schemes. Minimal effort, minimal risk of errors and maximum speed to deliver. In contrast, the NC/LC approach would require the administrator to click through every single bike component, select the new colour from a dropdown menu or with a radio button and confirm this selection. Obviously this route takes much longer, is prone to errors and it requires extra knowledge when colour schemes are desired.



A comparison of working steps: Changing the colour of a bicycle frame with a NC/LC platform (top) versus Infrastructure as Code (bottom).

Decision criteria

NC/LC appears easier approachable because of its visuality. However, the problem remains in the detail of the configurations that still need to be made.

IaC on the other hand appears very technical as it resembles expert thinking in its structured approach.

Here's how to make sense of the decision, depending on your context:

Here and there deployment or broader transformation?

When facing a single deployment, NC/LC impresses with its simple and quick handling. The drag-and-drop platform enables a fast implementation without prior knowledge and delivers quick outcomes.

But the ease of integration plays a major role in the IIoT deployment: Unlike other approaches, IaC smoothly integrates into any existing system and pipeline. There is no need to change the current infrastructure, which saves plenty of time and trouble. This also includes the potential of using different services applied on the same infrastructure - non-competitive and simultaneously. IaC also lowers the risk of flaws in processes of adaptation, as it naturally brings versioning. During the development, starting from the very first POC to the finalized environment, all steps stay traceable. Furthermore, staging deployment is easily applicable with IaC to test whether e.g. the system works properly before its final release. Additionally, a safe setup deployment can be integrated.

What's my outlook - a limited or scaling scenario?

A crucial advantage of IaC is its limitless scalability. IaC easily adapts to every expansion, enhancement and further development. Practically, new machines or whole heterogenous shop floors are quickly connected. Also, multiple IaC can be deployed and cross-connected, if a diverse handling internally or even externally across factories is required. What makes IaC even more beneficial and suitable for scaling is its capability

to design the data output individually. The commissioning files can include instructions on the frequency, format, content and critical threshold values of the data output.

In a limited use-case, NC/LC convinces with a time-saving deployment. It's highly specialized for the specific use case, which makes it straightforward and user-friendly.

What do I expect - Static setup or continuous improvement and automation?

Thanks to its standardized deployment, IaC reduces this risk of errors and false configurations which accompanies continuous improvements. More importantly, IaC enables full automation across multiple manufacturing processes, machines, locations and even different factory operators. Besides, it permits the implementation of specific parameters, if an individual adaptation is desired.

After all, if the need occurs to change the NC/LC infrastructure to a code-based infrastructure, the conversion can become time-consuming: As users are dependent on the pre-programmed underlying code of the interface, the transition might be intransparent and produce some unexpected results. Additionally, the predefined components rule out individual specifications. This choice hampers a factories' potential for adjustment to market changes and scalability.

Who's working with it - anyone and their grandma or experts?

Besides the obvious aspect of user-friendliness, the security of an IIoT environment is crucial. IaC scores with a highly secure infrastructure. By using commissioning files, a closed system is created, being naturally intangible for external access. If the latter is desired, pre-defined external access can be permitted explicitly within the code. This way, the privacy protection is ensured and the access stays limited to only intended requests.

NC/LC in contrast, stands out with its user-friendliness. Reduced to a graphical interface and as little code as possible, even non-experts can set the environment up or apply changes immediately if necessary.

What becomes apparent is that the IaC approach is more problem generic. It can be generalized easily to other environments and conditions. NC/LC in contrast, is more problem-specific and suitable for only one precise environment.

To accelerate your decision-making and summarize the main features of both approaches, you can rely on the following to rule of thumb:

IaC: Easy things are easy and complex things are possible.

NCLC: Easy things are really easy but complex things may not be expressible at all.

What's next?

For more than five years, Cybus has been relying on IaC to deploy an IIoT manufacturing environment. While realizing manifold solutions with diverse customers, our IaC solutions always contribute significantly to scalable, sustainable and efficient IIoT manufacturing environments.

If you have any questions or if you would like to get deeper into the topic of IaC, our experts are happy to provide you with further advice. You are welcome to contact us directly via hello@cybus.io!

References

Bitkom e.V.: Paulsen, N., & Eylers, K. (2020, May 19). Industrie 4.0 – so digital sind Deutschlands Fabriken.

Retrieved July 20, 2020, from <https://www.bitkom.org/Presse/Presseinformation/Industrie-40-so-digital-sind-Deutschlands-Fabriken>

PAC Deutschland: Vogt, Arnold (2020, April). Das Internet der Dinge im deutschen Mittelstand.

Retrieved July 21, 2020, from <https://iot.telekom.com/resource/blob/data/183656/e16e24c291368e1f6a75362f7f9d0fc0/das-internet-der-dinge-im-deutschen-mittelstand.pdf>

Cybus is a specialist for secure IIoT Edge software, headquartered in Germany. Cybus Connectware serves smart factories as a universal Edge and DevOps hub. Machine builders and providers of IIoT services use the Cybus Connectware as a software-based gateway. As early as 2017, Cybus published the first secure industrial connector for machine data according to today's DIN SPEC 27070 standard. Industry analyst Gartner named Cybus a worldwide „Cool Vendor“. Today, the company counts medium-sized and large companies from numerous industrial sectors such as mechanical engineering, automotive and aviation among its customers.

Cybus GmbH · Osterstraße 124 · 20255 Hamburg · Germany · www.cybus.io · hello@cybus.io · (+49) 40 228 58 68 51