The IoT Show delves into topics on Industrial Internet of Things (IIoT) and touches on the broader Internet of Things (IoT). We pick engaging, hot topics, and ask our speakers to advise on situations, opportunities, recommendations and gotchas so that listeners can benefit from the experiences and insights of others.

This document is not intended to be a complete representation of views of either the host or participants or a more formal white paper on topics discussed; it's more an aide memoire of comments made by show participants. To watch this episode in full, please go to this page.

In episode 2 we look at some of the basic questions on the role and value of the 'Platform' to those looking at the (industrial) internet. The host and three guests were:

Allan Behrens

MANAGING DIRECTOR AND

Bill Boswell

VP OF MARKETING, CLOUD

Diego Tamburini

PRINCIPAL INDUSTRY LEAD AZURE FOR

Dr. Josef Waltl

GLOBAL SEGMENT LEAD,

Before discussing the ‘Platform’, let’s try and level set. What does the industrial internet mean? It’s about data and ultimately insights. Insights being made from data in context, often in brownfield sites; helping to understand how products, machines and plants operate, their situation and their health. Also, how your customers are using products. Ultimately the IIoT allows you to competitively differentiate your business and/or products.

So, what is a, or the ‘Platform’ when speaking on IIoT/IoT?

For one thing, it’s a term often mired in confusion. There are many interpretations, depending on whom one talks to. In the context of *The IoT Show*, we’re talking of a software platform that delivers a set of software services; capabilities, if you will. These services (capabilities) are exposed (and made available) through API’s (Application Programming Interfaces) and are most likely available through the Cloud. Your own code stitched together with these API calls, forms the basis of your IIot/IoT application or solution.

Why would one need to consider an IIoT/IoT platform?

In the world of IIoT/IoT you’re dealing with complex networks of devices, machines, buildings, sensors and possibly actuators; often many tens of thousands of connections and data points, distributed geographically, perhaps across multiple continents. Data (and possibly control) from these sources needs to be streamed, processed and acted on. One needs to be mindful of security, perhaps across firewalls, and the availability and reliability of these connections and their data. One could potentially program all of this oneself, but the costs and effort would be prohibitive. Instead we use someone else’s ‘Platform’ on which we can easily build our (often unique) applications and capabilities.

IIoT/IoT Platforms deliver degrees of commonality across areas such as security, data streaming and analytics. So too, connection to new and brownfield (legacy) hardware via support of common protocols, such as OPC UA. The Platform may also deliver industry specific function, connection and certification, meaning that users need only spend their time working on areas that deliver clear differentiation for their customers.

Bear in mind that the Platform may not be just about what's on the Cloud. It may be about hybrid environments, supporting connections (across the Automation Pyramid) from lowest levels of device interface, to the highest level for instance to ERP; from real-time, perhaps analytics or other activities performed at the Edge, to speeds required at factory or corporation levels.

What should one think about when considering a Platform?

Firstly, consider the protocols you're interested in. OPC UA is a common one, but there are many that connect to legacy devices and manufacturing systems.

There will always be a mix of systems across vendors, both old and new. One must be able to ingest from your required sources. And this isn't limited to your devices and equipment. Think bigger, your ERP, PLM, SCM and other back office systems.

It's all about reducing the need for your company having to become experts on every aspect of (IIoT/IoT) programming. Leveraging the Platform's levels of granularity and ways to do specific industry things; domain knowledge/competence is important, and you might want to capitalize on that.

When working in the world of IIoT/IoT you might want to work at different levels of programming granularity, and the Platform should support (not hinder) that.

There is also significant value add to what partners can add to your developments, perhaps from their unique domain-centric knowledge, but also from tools and techniques, for instance in areas of analytics.

What can we learn from others in how to choose a Platform?

There are two threads of consideration and some common observations on how to determine what Platform is right for you. These are if you're an end-user and want to develop IIoT/IoT applications for your company. The second being if you're a software developer wanting to sell an application to customers. There are some common decision-making elements. These include:

- Reputation
- Availability of data centers

- Latency
- Regulations and standards
- Compliance
- Security
- Protocols
- Service level agreements
- Cost estimates for running your solution

For end-users, firstly decide whether your use case is so unique, you cannot possibly consider using an existing solution. It’s a build vs. buy decision. If you decide your application must be unique, have you the development skills, or at least the resources to hire them into your company?

Whether you’re developing for yourself or keen to use as much as possible from others, you may want to benefit from the velocity of innovation from different partners and providers. In this case, make sure the platform encourages 3rd-party involvement.

If you must develop for yourself, firstly consider if you can stitch together a solution from what’s available from existing Cloud Vendors to help optimize your efforts. Consider what Platform as a Service (PaaS) building blocks are available.

There are intermediate scenarios. These are usually described as Infrastructure as a Service (IaaS) and are generally vendor solutions that run on the Cloud on Virtual Machines (VM’s) or Cloud Provider Platforms. If you choose among these, pay note to whether solutions are properly certified on the VM’s, as well as the provider’s different levels of available VMs, and their portability. Beware of vendor lock-in and make sure what you’re developing is portable across different vendor’s VMs.

If you’re a software vendor developing an application for others, what is your core competency and value add. For instance, are you really an expert in AI? If not, these sorts of skills can be awfully expensive (and sometimes impossible) to add to your business. Are the building blocks available from the vendor, and will your customers mind what platform you'll chose? Finally, what platform gives you most flexibility, perhaps you might even consider offering platform support from multiple vendors?

At the end of the day there's no right or wrong. Consider how fast do you want to develop your solution vs your need to cater for specifics. It's much to do with opportunity cost vs. speed to market and at the end of the day it's about what your customer (internal or external) needs.

Bigger/Smaller business size and your choice of Platform

When considering a choice of platform, your company size may play a part in the decision. Larger companies may want to customize and extend; using partners or internal IT groups to deliver new applications. Others, perhaps smaller companies may want a more 'out-of-the-box' experience. Perhaps even by downloading applications from a e-store.

For the smaller business, perhaps you might want to consider the following advice:

- Start with (existing) applications already available for you to start with or build on.
- Make sure what you do is scalable (and global). Make sure scale-up doesn't incur unforeseen, additional charges.
- Start small, on something that isn't business critical but proves value.
- Have an end in mind. Perhaps a problem to solve and focus on proof of value, not proof of concept.
- Make sure that what you're aiming for is well defined.
- Create your solution using a small, motivated multi-disciplinary team
- Use available gateways (wherever possible) to ingest data from devices and equipment
- Only develop 'New' when you tried and succeed with the 'Old'.

Any last observations?

Connectivity is the base commodity delivered by any platform. Make sure whatever you chose is open, connects to the equipment or devices you need, supports multiple protocols and open formats.

Make sure that what you choose is secure, global, and simple to get started on. Partner applications and solution coverage helps to add value to the platform.

Is it worth it, and do you have the (development) skills to do it yourself?

"The closer you get to the Iron, the lower you get in the Platform Stack" – Meaning the closer you get to using 'bare metal servers', the more complex your programmatic access will be to (more granular) software subsystems.

> You might have more control over development, form and function, but you'll need to make architectural decisions such as what storage, where, or which ingestion systems etc.
>
> On the counter side, up in the stack means you have less control but this may remove many concerns.

This is exciting and even long-serving employees get enthused of the possibilities. Trust in the innovation of your team and don't wait; your competitors probably aren't.

It's not a matter of being small or large, more a matter of your appetite to develop your own solution and to what granularity. Remember that whatever you develop, you'll have to manage and support the finished product.