INTERNET O THINGS

DATA
VISUALIZATION
INPUT/OUTPUT
LOGIC

Edge Computing Primer

IoT Intelligence Starts at the Edge

OPTO 22
www.opto22.com

Opto 22

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: **800-321-6786** or **951-695-3000**
Pre-sales Engineering is free.
Product Support is free.
www.opto22.com

Edge Computing Primer

IoT Intelligence Starts at the Edge

On the train to work, Lee opened an email on her smartphone sent from a PAC (programmable automation controller) operating a surface-mount tool at her factory. The PAC attached a quality control report to the email that suggested changing the tool's solder temperature.

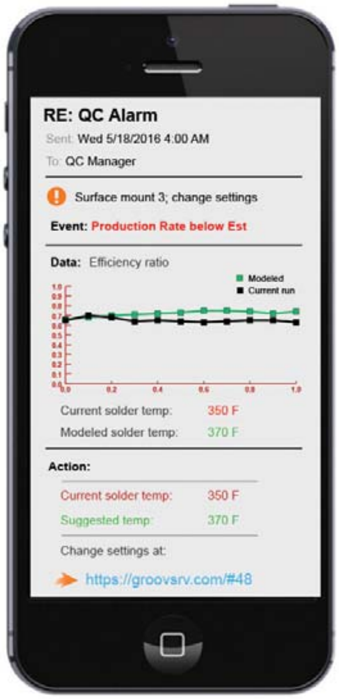
To generate that email suggestion, the PAC had securely sent yesterday's production data to a cloud-based analytics system to compare current and historical data for the machine.

Next, it accessed the machine manufacturer's website and obtained the latest recommended settings.

Finally, the PAC built a production efficiency report with a suggested solder temperature for today's production run that would increase yield by 7 percent over yesterday's run.

Lee clicked a link in the email and connected to the PAC's mobile interface over a secure, encrypted channel. Lee logged in and navigated to the machine's solder temperature setpoint, where she entered the recommended value.

All this took place before she got to the office.



RE: QC Alarm

Sent: Wed 5/18/2016 4:00 AM

To: QC Manager

Surface mount 3; change settings

Event: Production Rate below Est

Data: Efficiency ratio

Current solder temp: 350 F

Modeled solder temp: 370 F

Action:

Current solder temp: 350 F

Suggested temp: 370 F

Change settings at:

https://groovsrv.com/#48

PAC at the Edge

That PAC operating the surface-mount tool at Lee's factory operates at the edge of the factory's network.

Systems like these at the network edge are increasingly able to leverage cloud-based resources to perform edge computing—if computing resources exist as needed along the path from a sensor to the cloud—and if these computing resources reduce the total amount of data to be sent to the cloud for storage, processing, and analysis.

As a result, businesses can more quickly identify real opportunities for operational efficiency improvement and meaningful revenue generation.

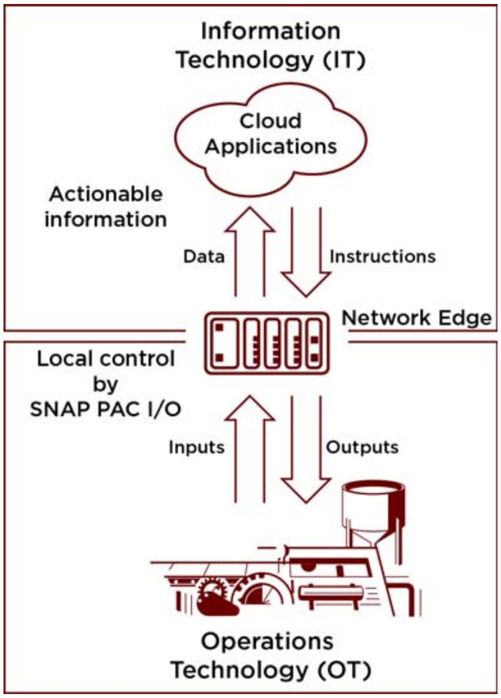
To foster such business benefits, data from the physical world of machines and equipment must be available to the digital world of the Internet and information technology systems, quickly, easily and continuously.

Successful IoT applications require operational technology (OT) professionals to make data from their systems, which monitor and control the physical world, accessible to the data processing systems of information technology (IT) professionals.

Once the data is there, cognitive prognostics algorithms running on IT systems can analyze it, refining raw physical data into actionable information that can predict outcomes in real time. The results can be used to improve inventory management and predictive maintenance and reduce asset downtime.

But before such benefits can be realized, three problems need to be solved: connectivity, big data, and IoT architecture.

The Connectivity Problem


Information Technology (IT)
Cloud Applications
Actionable information
Data
Instructions
Network Edge
Local control by SNAP PAC I/O
Inputs
Outputs
Operations Technology (OT)

The Internet of Things runs on vast amounts of data, generated by the physical world and then transported and analyzed by the digital world. It's an attempt to achieve perpetual connectivity and communication between people and things and even between things and other things.

But unfortunately most of these things were never designed to serve this new purpose. They were designed and installed long before the Internet was developed.

At the edge, things like sensors, circuits, relays, and meters are attached to industrial control systems used to operate equipment and machines. These sensors translate what's physically happening in the world (temperature, light, vibration, sound, motion, flow rate, and so on) into an electrical signal like voltage or current that can be interpreted by other systems to monitor and control physical equipment and machines.

These sensors typically have little or no intelligence and are designed to merely observe and report. They were not designed to communicate with the digital world of the IoT.

The physical world's language, the language of flow meters, temperature sensors, switches, and relays, is rarely digital bits and bytes. Their language is not the ones and zeros that information technology and Internet devices understand and use to communicate.

They also lack the physical connections and logical interfaces to communicate on the Internet of Things. They do not have a built-in Ethernet jack or wireless interface.

Few sensors and circuits can connect to the Internet, let alone speak or understand the languages the Internet uses, like JSON, RESTful APIs, and JavaScript. They don't run an operating system or have a built-in TCP/IP stack or web server.

And they have little or no built-in computing power, so providing edge computing at this level to filter volumes of data before forwarding to the cloud is impossible.

Right now the Internet and the things we want to connect to it aren't communicating. There's a disconnect between the physical world of current and voltage and the digital world of servers and clouds.

Integrating these disconnected things and systems is no small task. And with the significant potential technical pitfalls and risks of integrating these disconnected systems, we begin to wonder how long it will take to realize return on our investments in IoT applications.

One option is to simply wait for highly intelligent, connected sensors to become available to the marketplace. But those sensors are years away from being cost effective.

Moreover, sensors installed today or even decades ago are still performing their tasks. They're just not connected to the IoT, so the data they generate is siloed and inaccessible to IT systems for further analysis.

What is Edge Computing?

You've heard about **cloud computing**, which is using a network of remote servers, rather than a local server or personal computer, to store and manage data and run computer programs.

The advantages of cloud computing are that people or organizations can combine and share computer resources rather than having to build and maintain them.

Edge computing basically brings cloud computing down to the edge of the network, in the physical world. Edge computing is using computing power at the edge to filter or process data and then send only the required data to the cloud.

The advantages of edge computing are many. First, it reduces traffic on networks and the Internet by reducing the amount of data sent. It also plays a valuable role in efficiency, security, and compliance.

Right now the Internet and the things we want to connect to it aren't communicating. There's a disconnect between the physical world of current and voltage and the digital world of servers and clouds.

The Big Data Problem

Across the globe a massive installed base of things exists today, generating useful data that the IoT wants to access and consume. In oil and gas applications, a typical oilfield has up to 30,000 sensors installed. Factories and plants across the world have billions of sensors.

Each sensor is capable of generating huge amounts of data from the physical world. Some IoT applications could potentially generate terabytes of data per second.

These are volumes of data the digital world has never seen before. This is the Big Data problem.

Moving that much data onto existing network and Internet infrastructures for cloud-based analytics and centralized management will clog networks, vastly increasing network and Internet latency. For many industrial IoT applications, that is not acceptable, because real-time control and monitoring are mandatory.

For the Internet of Things to reach critical mass, intelligence must be pushed to the network edge, where the physical world meets the digital world.

Computing systems at the network edge must have the capability to collect, filter, and process data generated at the source, before it's transmitted up to the IoT.

And at the same time these edge computing systems must be able to complete the local real-time process control and automation tasks of traditional industrial applications.

The IoT Architecture Problem

Let's take a look at how today's IoT architecture works, so we can see its complexity and perhaps find a path forward.

For a cloud-based server to capture data from an analog sensor today, the sensor's data must be translated using a series of disparate software and hardware tools.

First, the sensor is physically wired to a device such as a PLC (programmable logic controller). While modern PLCs do provide basic analog-to-digital conversion of sensor signals, PLCs were not designed to interface with the Internet of Things.

PLC hardware, software, and programming languages were designed for repetitive, application-specific tasks like process control and discrete automation. They typically use proprietary protocols and languages for communication and programming, and do not include information security standards like encryption and authentication.

PLCs were originally designed as standalone systems. The protocols they use are seldom Internet compliant and are designed for point-to-point communication instead of the point-to-multipoint communication architecture found in the IoT ecosystem.

If systems that communicate using Internet-compliant protocols—such as PCs, web servers, and databases—want to communicate with a PLC, a vendor-specific and often proprietary software driver or hardware-based protocol gateway is required.

OPC (Open Platform Communication) software is one solution to this communication disconnect. But OPC was originally designed around PC architecture using the Microsoft Windows-only process exchange, COM/DCOM. Most systems and devices connecting to the IoT are not Windows-based devices.

For example, take your smartphone. It's likely an Apple or Android device, both of which run modified versions of the Linux operating system, where COM/DCOM process exchange does not exist.

OPC UA (Unified Architecture) has been released, but it's merely a wrapper for existing OPC drivers built on

What is an Input/Output (I/O) System?

Things in the physical world typically communicate in an electrical signal like voltage or current, but the digital world doesn't understand these signals. I/O systems act as the translator.

I/O systems are wired directly to sensors and actuators. Their job is to convert analog data (like voltage or current) from a sensor into digital data (ones and zeros) to send to a computer, or to convert digital data into analog data to send to an actuator, to take action in the physical world.

An input sends data from the physical thing to the computer; an output does the reverse. Inputs and outputs are like the fingers on a hand, sensing the world, sending data about it to the brain, and then taking commands from the brain back to the fingers to take an action.

Windows architecture. It requires design engineers to build an OPC UA client adapter into their products. And even then, modern network and Internet assets such as web servers, databases, smartphones, and tablets do not speak OPC UA.

PLCs, OPC servers, proprietary drivers, and protocol gateways quickly become a convoluted IoT architecture. These layers of complexity not only require time, money, and specific domain expertise to install and maintain, but also the data being sent from the physical world has been converted by so many different pieces of hardware and software that data integrity can be jeopardized.

Imagine the difficulty in provisioning and troubleshooting these IoT systems.

And then consider that today's automation architectures often do not address information security. Sending data generated at the edge through so many layers of conversion not only increases network latency but also opens up complex information security concerns as the data is transported to the cloud.

Multiply these issues across the billions of devices we expect to connect using the IoT, and you see the communication challenge the IoT faces.

There has to be a better way.

Flattening the IoT Architecture

As we've seen, for the IoT to reach critical mass, Internet protocols and technologies need to be driven into systems at the edge, where the physical world and the digital world connect.

Layers of complexity must be removed from the communication process between digital systems and physical assets. Modern IoT system architectures must be flattened, streamlined, optimized, and secured.

If we drive Internet connectivity and data processing power into edge devices, we can greatly accelerate our time to insight and action. Edge computing devices will become the sensor on-ramp for the billions of data points we intend to connect to the IoT.

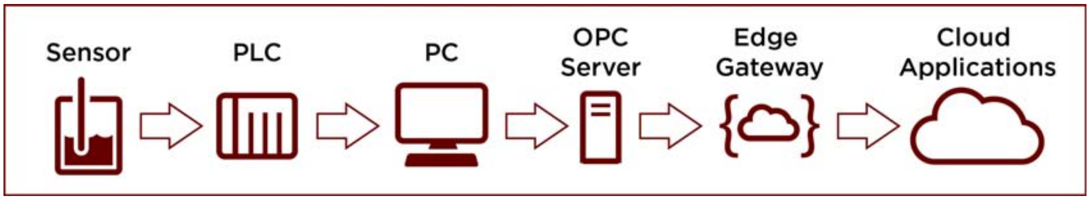
These edge computing systems will need the ability to receive the input signals of the physical world and output the meaningful data the IoT needs, in a form that digital Internet-enabled systems already understand.

Edge computing systems must easily and securely access the cloud through the open, standards-based communication technologies the Internet is based on.

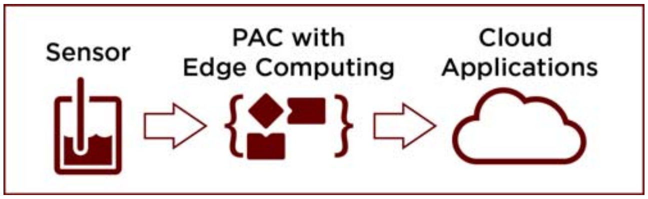
That means:

- **Internet technologies** like TCP/IP, HTTP/S, MQTT, and RESTful APIs—the dialect of the Internet—must be built directly into the input/output level, or the point of physical to digital conversion.
- Internet **security** technologies like SSL/TLS encryption and authentication must be built in directly to edge computing systems.
- Cloud-based systems must be able to make **RESTful API** calls to access data, or subscribe to data points on remote edge devices, without the layers of complexity and conversions that exist in industrial applications today.

The Problem: Complex Current IoT Architecture



The Solution: Flattened IoT Architecture



The Power of Interoperability

We did not always have one cohesive system for sending and transmitting information. Before the Internet and the world wide web, many different Internet-like protocols and architectures existed. Computer systems all ran different operating systems requiring different programming languages.

Small pockets of interconnectivity existed, but for the most part systems were disconnected from each other. It was very similar to the way industrial systems communicate today, with the need for converters, adapters, and gateways.

The Internet was designed to allow input/output and information systems to share data through a common interface, removing layers of complexity and allowing for greater interoperability between systems designed and manufactured by different vendors.

That's why an Apple computer or Android phone today can send an email to a Windows computer: they speak the same Internet languages. Today's Internet uses a common set of protocols, tools, and routines designed to make the transportation, acquisition, and analysis of digital information a seamless process, no matter what device you're using.

Although sensors and other physical assets installed at the edge may not have been designed with Internet interoperability in mind, there's still a massive opportunity to collect meaningful data from the huge installed base of existing things.

But it will require a solution that understands both sides of the OT and IT convergence—something that can:

- Locally **translate the physical world** of currents and voltages (OT) into the secure, RESTful APIs and JSON (JavaScript Object Notation) frames the digital world (IT) understands
- **Process and filter mountains of data**, sending only the necessary data to the cloud for analysis
- Provide communications interfaces and processing power to **maintain the closed-loop, real-time control requirements** of industrial applications.
- Deliver all of the above in a package **suitable for challenging industrial environments** where dust, moisture, vibration, electro-mechanical frequencies, and temperature vary widely.

In the long run, OT/IT convergence will demand a flattened architecture and seamless communication between assets, using open, standards-based communication protocols and programming languages.

Conclusion

We've seen that edge computing is the sensor on-ramp to the IoT. Until the communication, security, and computing technologies of the Internet find their way into computing at the edge, the IoT will fall short of its potential.

Internet technologies are available in some industrial systems today. And some vendors have already started bridging the gap between OT and IT by adding IoT technology like MQTT, RESTful APIs, and JavaScript directly into programmable automation controllers (PACs).

Our shortest path to a successful IoT is to leverage the existing interoperability technologies of the Internet in industrial automation products and applications.

How can Opto 22 help you?

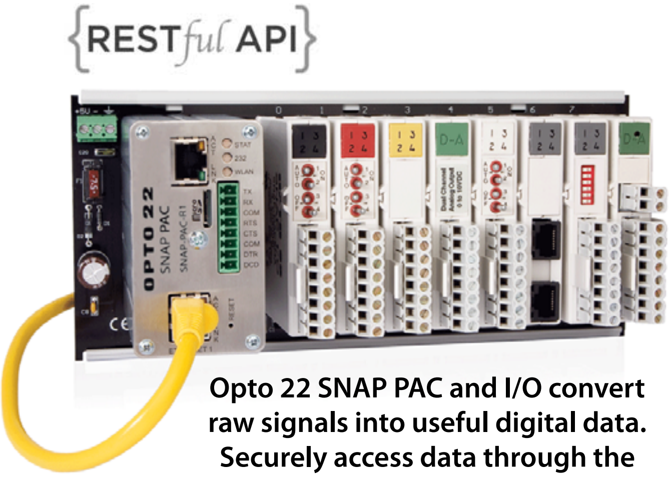
At Opto 22, our goal is to flatten the IoT architecture so you can realize your IoT goals.

Opto 22's engineering focus is on building hardware and software tools to bring the benefits of the IoT—simply, reliably, and securely—to the things that already exist in your world.

Our PACs offer an easy and cost-effective way to bridge the real world with the digital world, through a comprehensive collection of input and output (I/O) modules designed to connect with virtually any electrical, electronic, mechanical, or environmental device.

This I/O system converts these raw signals to useful digital data and shares it over the standard networks and protocols understood by IT.

For edge computing, our SNAP PAC controllers offer field-proven control, data collection, and logic solving—plus a RESTful API so you can access that data using the Internet programming languages you're familiar with: .NET, PHP, Perl, JavaScript, and more.



Opto 22 SNAP PAC and I/O convert raw signals into useful digital data. Securely access data through the PAC's HTTPS server and RESTful API.

When it comes time to visualize, notify, and mobilize your information, our *groov* platform offers a simple way to build mobile operator interfaces—mobile apps to securely monitor and control virtually any automation system or equipment. These interfaces can be used on any screen, from your smartphone to big-screen HDTV.

All Opto 22 products are backed by decades of expertise in applications like process control, discrete manufacturing, remote telemetry, data acquisition, and supervisory control.

Opto 22 products are supported by our experienced engineers at no charge and available worldwide.

About Opto 22

Opto 22 was started in 1974 by a co-inventor of the solid-state relay (SSR), who discovered a way to make SSRs more reliable.

For over 40 years, we've brought commercial, off-the-shelf technologies to industrial systems all over the world, designing our products on open standards. We pioneered the use of PCs in controls back in the 1980s, Ethernet networking at the I/O level in the 1990s, and machine-to-machine connectivity in the 2000s. Today, we build secure Internet technologies into our PACs and I/O.

All Opto 22 products are manufactured and supported in the U.S.A. Because the company builds and tests its own products, most solid-state SSRs and I/O modules are guaranteed for life.

MADE IN THE USA

For more information, visit opto22.com or contact **Opto 22 Pre-Sales Engineering**:
800-321-6786 or **951-695-3000**

Or email: systemseng@opto22.com

Opto 22 *groov* for building and viewing mobile operator interfaces