March 2021

INTENT-BASED NETWORKING FROM THE IOT EDGE TO THE APPLICATION SERVER

Tim Szigeti

Robert Barton

Jerome Henry

Dave Zacks

INTENT-BASED NETWORKING FROM THE IOT EDGE TO THE APPLICATION SERVER

AUTHORS:
Tim Szigeti
Robert Barton
Jerome Henry
Dave Zacks

ABSTRACT

The dynamic management of traffic within an operational technology (OT) network raises a number of challenges. To address those types of challenges, techniques are presented herein that enable end-to-end intent-based networking to control access between the OT domain and on-premise or cloud-based data center (DC) domains. Aspects of the presented techniques employ deep packet inspection (DPI) of industrial protocols within the OT domain (e.g. by sensors) and map Internet of Things (IoT) devices and traffic flows to abstract tags (through, e.g. a robust security facility), export such tags to a common policy server that bridges both domains, assign the IoT devices to corresponding security profiles (e.g., based on their device characteristics as expressed by tag metadata), and map the security profiles to specific fabric overlay microsegments (e.g., endpoint groups (EPGs)) within a DC or cloud domain.

DETAILED DESCRIPTION

Micro or macro segmentation policies can restrict traffic within an operational technology (OT) network (e.g., a cell or area zones in the Purdue model). However, some flows may need access to resources that are beyond the OT network, such as can be found in, for example, on-premise industrial or enterprise data centers (DCs), cloud-based DCs, etc. Segmentation policies cannot provide or control the access requirements between such domains.

For example, a robot on a factory floor may be required to upload the number of units that it has produced during the past hour to an ordering application in a DC to confirm what may be available to ship to customers. Another industrial device may need to communicate with, for example, a supervisory control and data acquisition (SCADA) system, an engineering management system, a historian, a data messaging system (such as a Message Queuing Telemetry Transport (MQTT) broker), etc. As such, there is a pressing need for scalable and flexible policy controls for end-to-end flows, such as, for example, from Internet of Things (IoT) endpoints all of the way to specific application servers.

Today, most access control policies are still administered through Internet Protocol (IP) access control lists. However, when a device appears on an industrial network – where, for example, device certificates and authentication mechanisms (based on, for example, Institute of Electrical and Electronics Engineers (IEEE) standard 802.1X) are rare – it may show up simply as a new media access control (MAC) address or a new IP address, with little additional detail provided to an administrator so that he or she can know which access policies need to be applied. Aspects of this are illustrated in Figure 1, below.

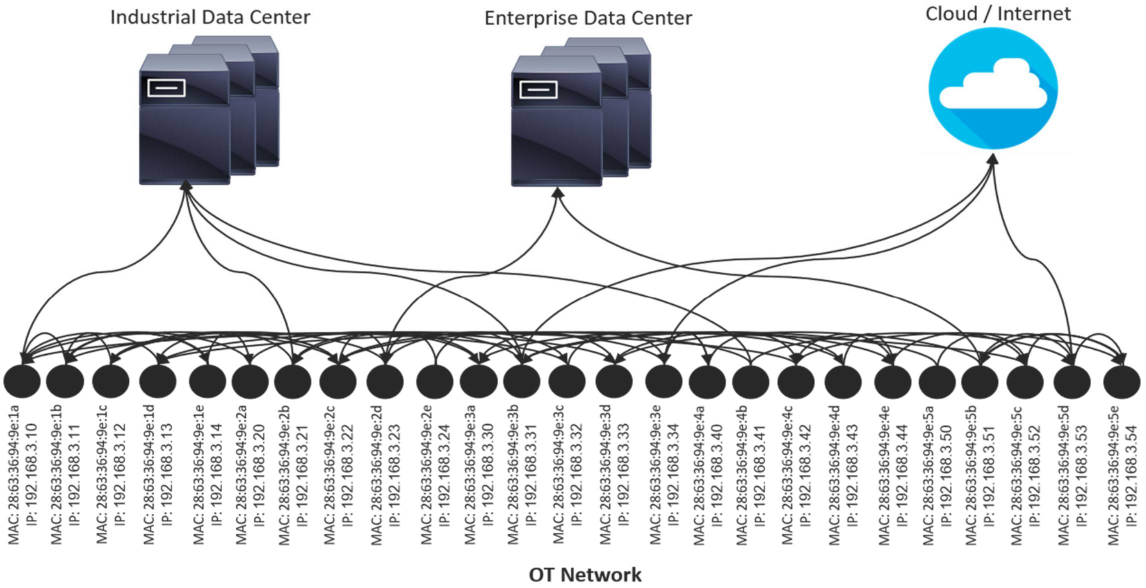

Industrial Data Center
Enterprise Data Center
Cloud / Internet
MAC: 28:63:36:94:9e:1a IP: 192.168.3.10
MAC: 28:63:36:94:9e:1b IP: 192.168.3.11
MAC: 28:63:36:94:9e:1c IP: 192.168.3.12
MAC: 28:63:36:94:9e:1d IP: 192.168.3.13
MAC: 28:63:36:94:9e:1e IP: 192.168.3.14
MAC: 28:63:36:94:9e:2a IP: 192.168.3.20
MAC: 28:63:36:94:9e:2b IP: 192.168.3.21
MAC: 28:63:36:94:9e:2c IP: 192.168.3.22
MAC: 28:63:36:94:9e:2d IP: 192.168.3.23
MAC: 28:63:36:94:9e:2e IP: 192.168.3.24
MAC: 28:63:36:94:9e:3a IP: 192.168.3.30
MAC: 28:63:36:94:9e:3b IP: 192.168.3.31
MAC: 28:63:36:94:9e:3c IP: 192.168.3.32
MAC: 28:63:36:94:9e:3d IP: 192.168.3.33
MAC: 28:63:36:94:9e:3e IP: 192.168.3.34
MAC: 28:63:36:94:9e:4a IP: 192.168.3.40
MAC: 28:63:36:94:9e:4b IP: 192.168.3.41
MAC: 28:63:36:94:9e:4c IP: 192.168.3.42
MAC: 28:63:36:94:9e:4d IP: 192.168.3.43
MAC: 28:63:36:94:9e:4e IP: 192.168.3.44
MAC: 28:63:36:94:9e:5a IP: 192.168.3.50
MAC: 28:63:36:94:9e:5b IP: 192.168.3.51
MAC: 28:63:36:94:9e:5c IP: 192.168.3.52
MAC: 28:63:36:94:9e:5d IP: 192.168.3.53
MAC: 28:63:36:94:9e:5e IP: 192.168.3.54
OT Network

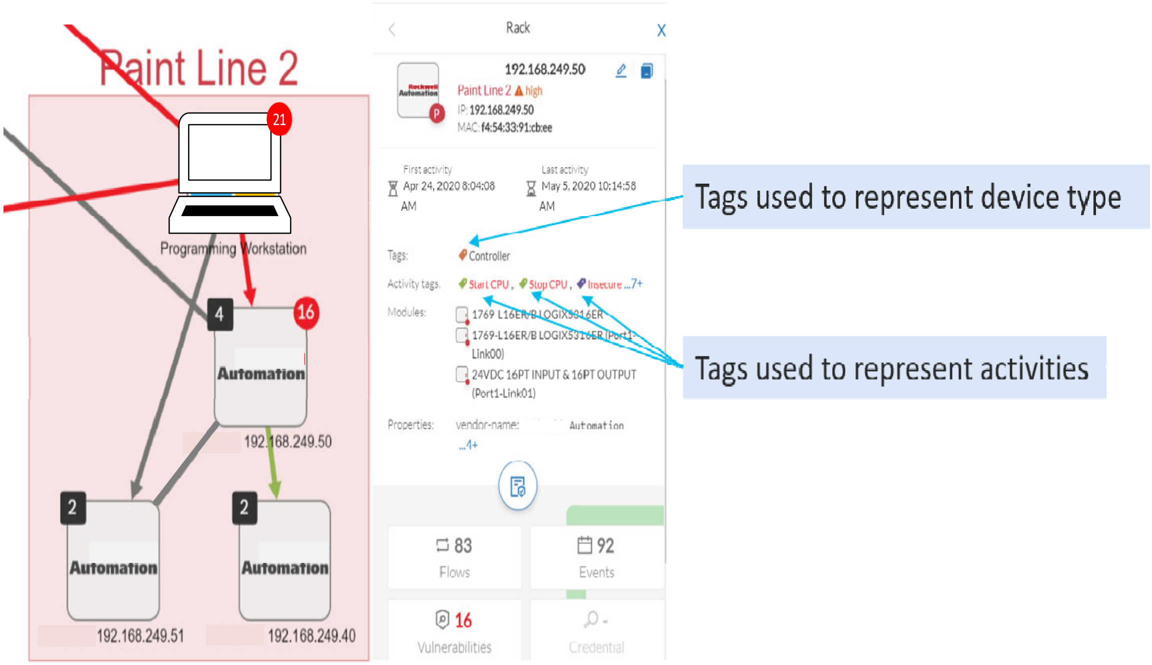
Figure 1: Illustrative Industrial Network

Even if a device is known, IP access lists are difficult to scale, maintain, troubleshoot, etc. and may be overly-restrictive, especially in dynamic environments – which applies to both OT networks (where, for example, devices are routinely hot-swapped) as well as to DCs (where, for example, compute resources are continually being redistributed and reallocated to maximize efficiency).

A secondary, and directly related, challenge is that operational data flows from IoT endpoints to application servers may represent high or low priority traffic. As such, administrators may require that these flows be correspondingly prioritized or deprioritized across the network and compute resources. For example, IoT endpoints may send failure or imminent-failure messages to a specific application in an industrial DC (e.g., in an Industrial Demilitarized Zone (IDMZ)). These messages may be prioritized over both the network and compute fabric to ensure that they are received and acted upon with urgency. Or, as an alternate example, some IoT endpoints may send non-urgent operational data for diagnostic or trending purposes, which may be deemed low priority. Such traffic may be de-prioritized across the network and compute fabrics.

Previous attempts at addressing the types of challenges that were described above have, among other things, focused on the concept of IoT device abstraction via tags. Such approaches may be used to integrate with the security domain, as well as to segment the OT network. However, segmentation policies define either the boundaries of the OT network itself (macro-segmentation) or the specific rules of intra-network communication within the OT network (micro-segmentation). As such, segmentation policies cannot provide or control access between domains outside of the OT network (such as between the OT network and a DC, whether on-premise or in a cloud).

Under one previous attempt at addressing the types of challenges that were described above, metadata about an IoT device is collected by sensors. The collected metadata may include, for example, IP addresses, MAC addresses, vendor details, device type, firmware version, the switch identifier and the port number where the device is connected, etc. As an edge sensor learns details of a new device, those details are sent as metadata about each device, and are associated with abstract tags in an asset inventory service. The inventory service examines all of the metadata information and tags for the new device that is being added. As new devices are discovered, the particulars are added to the live inventory. The tags are used to classify and describe the device, as depicted in Figure 2, below.



Rack

192.168.249.50
Paint Line 2 ▲ high
IP: 192.168.249.50
MAC: f4:54:33:91:cb:ee

First activity: Apr 24, 2020 8:04:08 AM
Last activity: May 5, 2020 10:14:58 AM

Tags: Controller
Activity tags: Start CPU , Stop CPU , Insecure ...7+
Modules: 1769 L16ER/B LOGIX5316ER
1769-L16ER/B LOGIX5316ER (Port1-Link00)
24VDC 16PT INPUT & 16PT OUTPUT (Port1-Link01)
Properties: vendor-name: Automation ...4+

83 Flows
92 Events
16 Vulnerabilities
- Credential

Tags used to represent device type

Tags used to represent activities

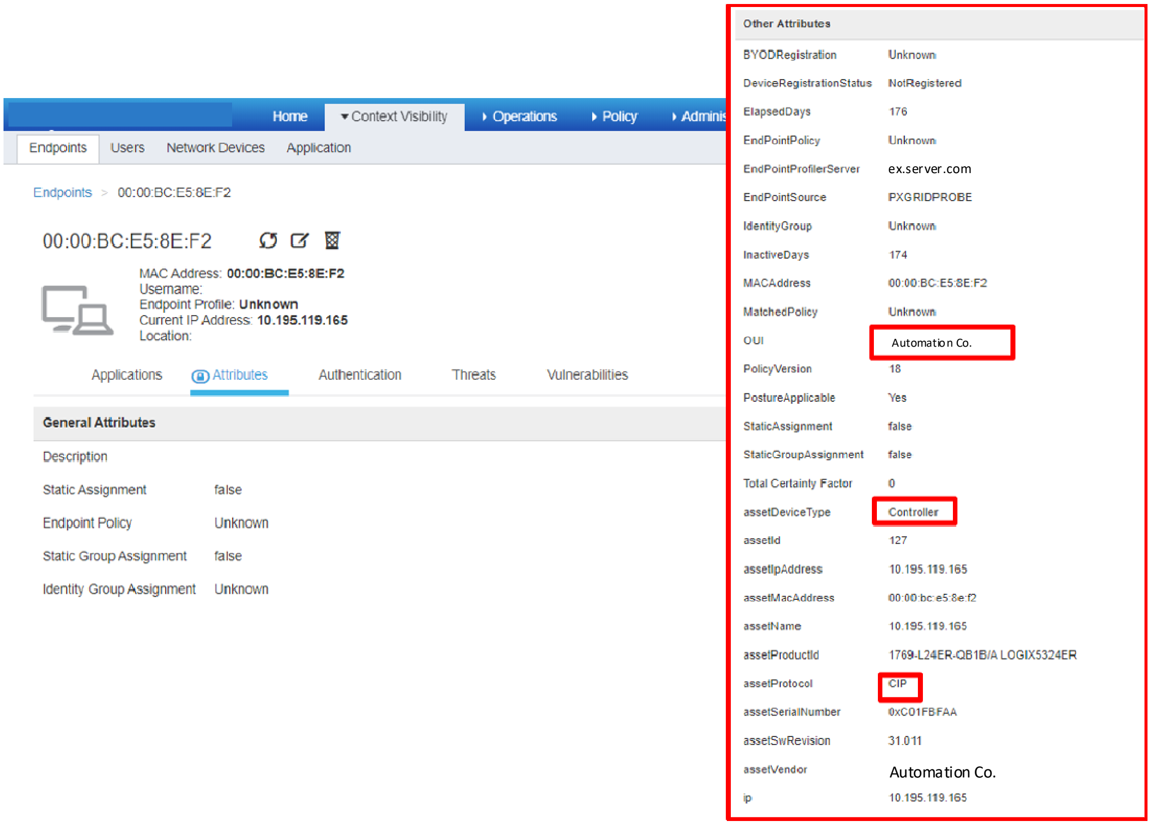
Figure 2: Exemplary IoT Devices Associated with Tags in Inventory System

However, these tags have no relevance or meaning outside of the OT domain. Furthermore, in external domains, such as a DC, there is no awareness of IoT device types, OT protocols, etc. As such, for intent-based policies to be expressed and enforced end-to-end, there must be a means to translate abstract constructs in one domain to familiar constructs in the other, and vice versa.

Another previous attempt at addressing the types of challenges that were described above employs a virtualized path identifier (VPR). The identifier is an integer that represents an abstraction of the route and destination (instead of a protocol-specific, formal IP destination and next hop sequence). Such an approach is useful to construct routes through heterogenous networks. However, the approach does not examine the notion of an extensible label that describes what a device is, the group that it belongs to, and what traffic it sends. It also does not examine the creation of access rules based on such labels.

To address the types of challenges that were described above, and to address the deficiencies that are associated with the various previous attempts that were described above, techniques are presented herein that support an intent-based approach for policy abstraction and enforcement that facilitates, among other things, inter-domain access control.

Under aspects of the techniques presented herein, device metadata may be shared from the OT domain to a common element that bridges the OT and DC domains such as, for example, a common policy server. This may be done through, for example, secure application programming interfaces (APIs), a security product integration framework, or other compatible vehicles. Aspects of the results of such an exchange are depicted in Figure 3, below.



00:00:BC:E5:8E:F2

MAC Address: **00:00:BC:E5:8E:F2**
Username:
Endpoint Profile: **Unknown**
Current IP Address: **10.195.119.165**
Location:

General Attributes	
Description	
Static Assignment	false
Endpoint Policy	Unknown
Static Group Assignment	false
Identity Group Assignment	Unknown

Other Attributes	
BYODRegistration	Unknown
DeviceRegistrationStatus	NotRegistered
ElapsedDays	176
EndPointPolicy	Unknown
EndPointProfilerServer	ex.server.com
EndPointSource	PXGRIDPROBE
IdentityGroup	Unknown
InactiveDays	174
MACAddress	00:00:BC:E5:8E:F2
MatchedPolicy	Unknown
OUI	Automation Co.
PolicyVersion	18
PostureApplicable	Yes
StaticAssignment	false
StaticGroupAssignment	false
Total Certainty Factor	0
assetDeviceType	Controller
assetId	127
assetIpAddress	10.195.119.165
assetMacAddress	00:00:bc:e5:8e:f2
assetName	10.195.119.165
assetProductId	1769-L24ER-QB1B/A LOGIX5324ER
assetProtocol	CIP
assetSerialNumber	0xC01FBFAA
assetSwRevision	31.011
assetVendor	Automation Co.
ip	10.195.119.165

Figure 3: Exemplary Exported IoT Device Attributes

After it is shared, device metadata may be used to gain access to a cloud or DC overlay segment in another network. For example, a Software-Defined Networking (SDN) or application centric networking solution may employ a logical separation of applications (microsegments) to isolate application behavior and to protect the applications. With many modern user-centric applications, a user gains access to an application through an API, user authentication, some multi-factor authentication (MFA) mechanism, etc. However, IoT devices (typically headless) do not support a mechanism to self-authenticate to an application and very rarely do they support API connectivity. These devices can trigger a connection, but cannot undergo the complex identity validation exchanges. Thus, aspects of the techniques presented herein employ device identity and activity tags learned by the network as a descriptive authentication mechanism that allows IoT devices to communicate with the correct application microsegment in the cloud or DC.

In the context of an SDN or application centric networking solution, applications are segmented by a SDN controller (e.g., a policy infrastructure controller) and communication within the microsegment makes use of an endpoint group (EPG) tag on the IP packet. At the edge of such a fabric, a gateway examines the credentials of the incoming packet flow and either grants it access to the microsegment (and gives it an EPG) or it denies the flow.

Under aspects of the techniques presented herein, the fabric gateway must associate an access policy to each microsegment of the SDN solution fabric (essentially, it must associate an access policy to the overlay segments of the fabric). Rather than use normal Transmission Control Protocol (TCP) 5-tuples to grant or restrict access, the incoming flow is compared with the known set of inventory tags associated with the device of origin. A security policy is constructed on the gateway device that only allows communication into a fabric microsegment that has the correct set (or a minimum set) of tags.

For example, an industrial programmable logic controller (PLC) may need to record an activity with a (e.g., Open Systems Interconnection (OSI)-based) process information (PI) historian server in the IDMZ which uses an application centric infrastructure fabric. A security facility, as described below, will have learned the tags of the device. When the PLC attempts to communicate with the application centric infrastructure fabric gateway, the set of tags for this device is examined. The activity tags are also compared to determine if this device is known to communicate with the process information historian server. If a minimal set of acceptable tags is present, the connection is allowed into the application centric infrastructure fabric overlay, and the correct EPG is mapped to the packet to ensure that it is secured.

Aspects of the techniques presented herein leverage, among other things, a robust security facility. Such a security facility may support, possibly among other things, the dynamic management of asset inventory and the real-time monitoring of process data. Additionally, such a facility may offer fingerprint recognition and anomaly detection capabilities. For example, from a set of known templates (as a starting point), Deep Packet Inspection (DPI) may be employed to examine each source, destination, and packet to distinguish normal from abnormal activity. Such activities may include multiple mechanisms, including asset or device type detection (e.g., connection point types, energy and transmissions structure, protocols, packet types, etc. may contribute to identity tags – i.e., what a device is) as well as activity detection (e.g., the development of activity tags – i.e., what a device is doing).

Thus, just like it is legitimate to expect a firewall to distinguish allowed from disallowed packets, it is legitimate to expect a security facility, as described above, to distinguish valid from invalid flows and packets (with the same caveats regarding misconfigurations, Common Vulnerabilities and Exposures (CVE), etc.). From this validation, tags or labels may be created and exported. From the labels, access rulesets may be built. As such, building on the principle that DPI can effectively analyze traffic and distinguish valid from invalid senders and packets, aspects of the techniques presented herein leverage this outcome to create access rules.

Within a security facility electronic tags may also have the capability to identify the security posture of a device. Within the application centric infrastructure fabric, a fabric overlay may contain sensitive applications that are vulnerable to certain types of attack from unsecure IoT devices (especially ones that are unpatched). The security policy of the application centric infrastructure fabric gateway is enhanced to correlate the security posture of the IoT devices with the known vulnerabilities of applications within the target overlay. If the IoT device or activity tags indicate a weak security posture on a known device, entry to the overlay can be restricted or controlled (e.g., in a tunable manner, depending upon the severity of the security threat).

Once mapped to an application centric infrastructure fabric overlay, traffic to or from the overlay can be controlled at the borders of the DC through traffic contracts that have been established between EPGs (i.e., a microsegment). Additionally, these traffic contracts can also establish service levels that the flows are to receive over the DC network and compute fabrics. As such, the traffic contracts can prioritize or deprioritize flows over both the network and compute fabrics according to administratively defined policies. Illustrative EPGs and traffic contracts are depicted in Figure 4, below.

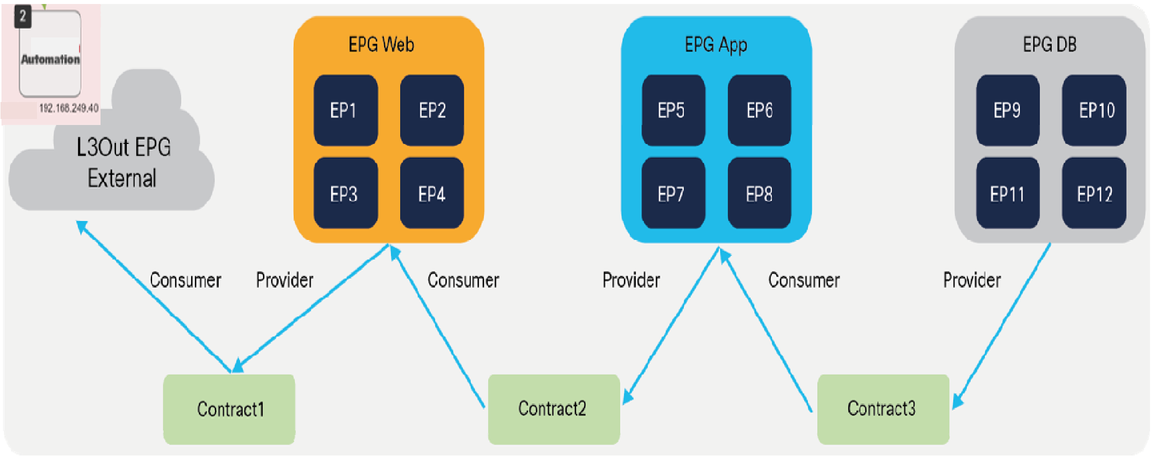


Figure 4: Illustrative EPGs and Traffic Contracts

Not only does such a method provide end-to-end intent-based security policy enforcement, but it also enables end-to-end visibility, monitoring, and assurance. For example, network analytics engines in both the OT and DC domains can now correlate traffic flows in their respective domains and can report their combined observations to IT or OT operators for an end-to-end view of traffic from IoT edge devices all the way to the application servers. Additionally, once a flow is allowed further connection to the IoT application may require an authentication choreography. As granter of the access authorization, the application centric infrastructure fabric gateway also acts as an authentication proxy. As the IoT device attempts to establish the connection the application centric infrastructure fabric gateway identifies authentication challenges that are returned by the application server. The gateway intercepts those calls and responds, thus authenticating in the name of the IoT object. Once the process completes, the gateway forwards the IoT traffic within the secure connection thus created.

It is important to note that existing access policy security tags typically assume a standard IT structured network, where tags are used to allow or block device-to-device (or network-to-network) communication. However, in an OT network such a structure does not yet exist. In most cases, OT traffic needs to flow to the IDMZ, to the cloud, and in some cases to the shared DC in the IT domain, and then back again. Aspects of the techniques presented herein support a systematic translation and rule creation structure that allows leveraging the elements that are native to OT, converting them into labels, then using such labels to create rules that can be applied in the IT space. In that space, the rules can be translated into the creation of micro-segments (e.g., with the associated access policy security tags or end point groups). However, access policy security tags alone may not address all of the challenges that were described above as a core missing element is the translation of OT identity and activity (e.g., security facility tags) into IT-leverageable labels and rules.

In summary, techniques have been presented that that enable end-to-end intent-based networking to control access between the OT domain and on-premise or cloud-based DC domains. Aspects of the presented techniques employ DPI of industrial protocols within the OT domain (e.g. by sensors) and map IoT devices and traffic flows to abstract tags (through, e.g. a robust security facility), export such tags to a common policy server that bridges both domains, assign the IoT devices to corresponding security profiles (e.g., based on their device characteristics as expressed by tag metadata), and map the security profiles to specific fabric overlay microsegments (e.g., EPGs) within a DC or cloud domain.

Further, traffic contracts are defined and enforced between EPGs for both access and treatment policies, allowing such traffic contracts to also apply to flows sourced-from or destined-to specific IoT devices. As a result, administrators no longer need to use cumbersome, unscalable, and error-prone IP access lists to enforce inter-domain access policies. Additionally, end-to-end visibility is enabled as the flows may be logically identified in each domain via domain-specific abstractions (specifically, through tags in the OT domain and EPGs in the DC domain) and correlation of these complimentary views may be done by applying the tag-to-EPG mappings performed at the central policy server.