VIXTERA TECHNOLOGY INNOVATION

VIXTERA DECLARATIVE PROTOCOL FACILITATES HIGH VELOCITY OF DEVICE ON-BOARDING, MAITENANCE AND MANAGEMENT

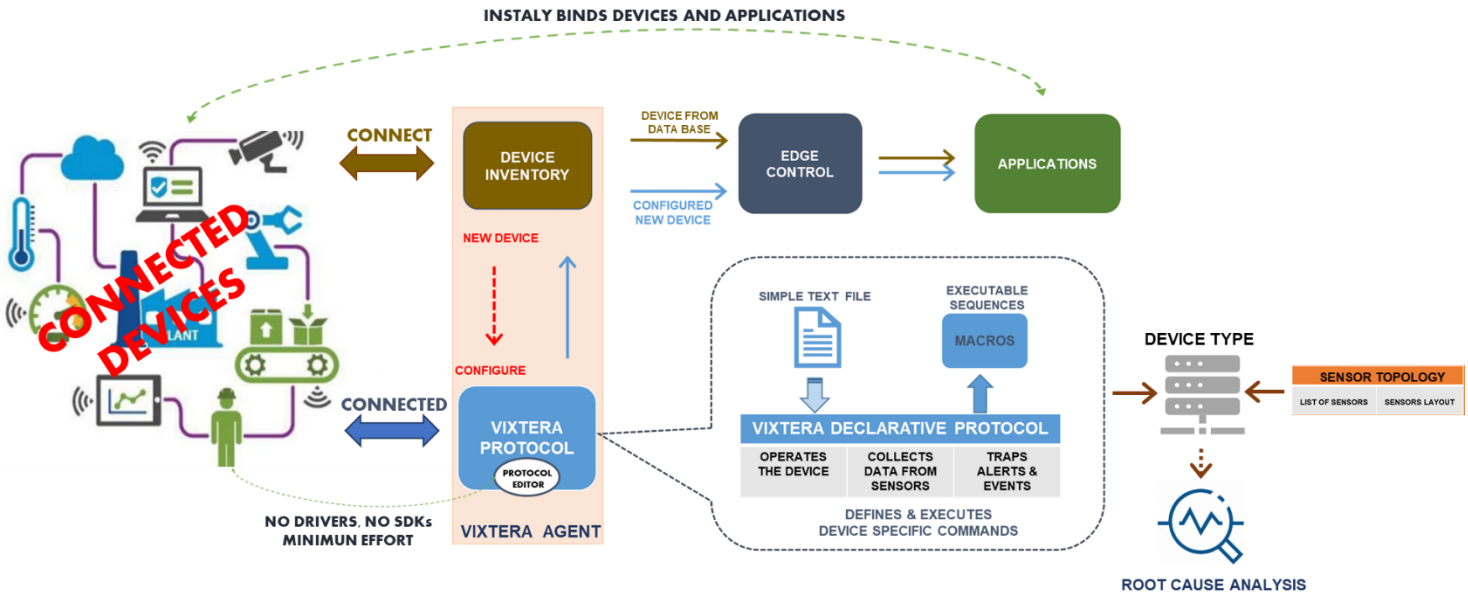
Any physical object can be transformed into an IIoT device..., if it can be connected to the Internet.

However, to date, there is no industry standard that define a common set of operations that can be executed by devices. Each device manufacturer develops its own protocol to define a set of supported operations for device control and data query. Furthermore, in order to enable software system to interact with devices, a specialized software and protocol operation has to be developed for each specific device and model.

Vixtera invented a methodology and filed a patent for software system/device interaction that assumes that all device protocols can be defined as a set of operations, and that any operation has the identical structure that is DEVICE AGNOSTIC.

Each operation can be defined declaratively using a common declarative language (XML, JSON) and can be interpreted by a driver common to all operations of all protocols, thus avoiding laborious processes of writing specialized software.

Several operations can be combined into sequences (macros) allowing rapid response to frequent changes and upgrades, while instantly adapting to ever-changing IIoT environment.



Everyday annoying problems and use cases easily mitigated by Vixtera's declarative protocol

Use Case #1

A customer deployed 150+ switches in its data center. Due to firmware issue, some of the switching devices have been sending up to 100 connect/disconnect events several times a week causing tremendous system overload. DC operator had to manually restart

devices for almost 6 months until the issue has been finally identified and fixed by the switch manufacturer.

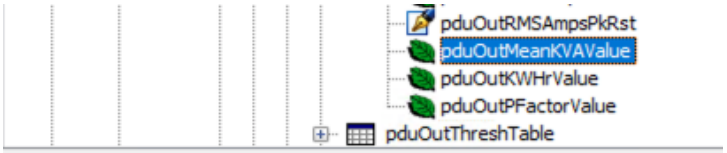
Vixtera's declarative protocol and technique enables instant and automatic execution of a devices' restart/reboot command based on a preconfigured sequence of events coming from this device.

Use Case #2:

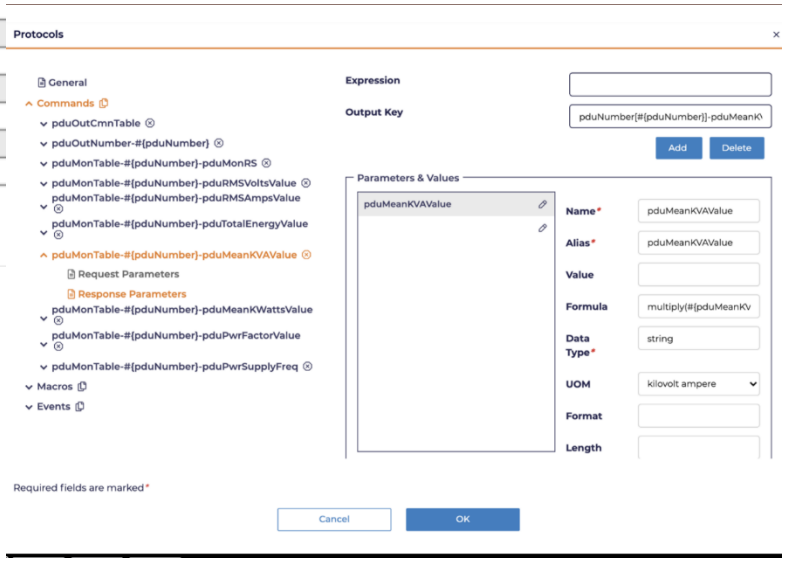
In many cases, the spec provided by device's vendor does not correlate to the information depicted from real data collected from this device in the field. For example, SNMP configured devices can send fractional data that would be perceived as a whole number and declares a scale (a discrepancy between Syntax that defines a type of this value and Description that mandate to treat this value as a multiplier) in the MIB file.

Vixtera declarative protocol's flexible model provides for easy change of the MIB file configuration by passing a simple formula for a specific parameter in the protocol while correcting any mis-calculation.

Example: "formula": "multiply(#{pduRMSAmpsValue}, 0.1)"



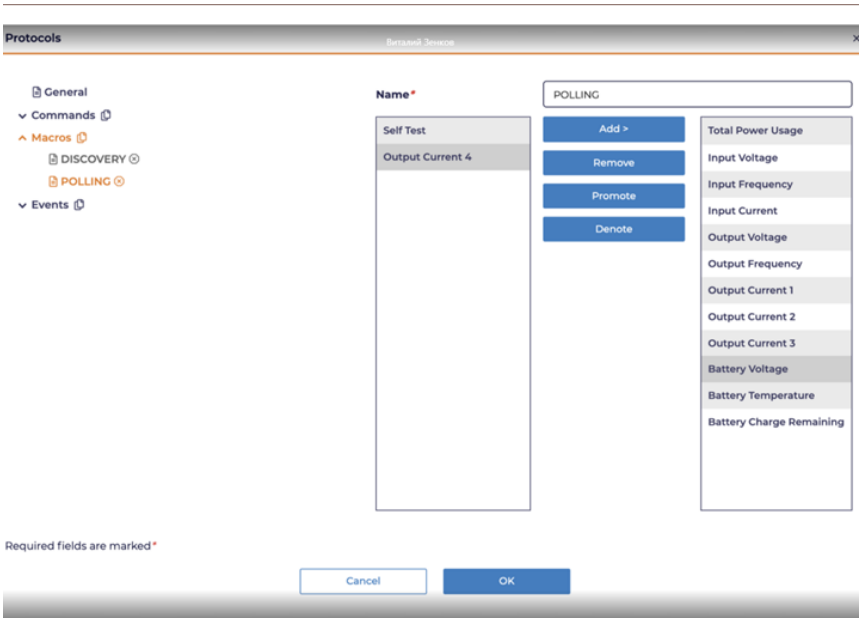
Name	pduOutMeanKVAValue
OID	.1.3.6.1.4.1.3711.24.1.1.7.2.3.1.13
MIB	HAWK-I2-MIB
Syntax	UNSIGNED32
Access	read-only
Status	current
DefVal	
Indexes	pduOutPduNumber, pduOutNumber
Descr	Mean kVA value at the outlet in 0.1 kVA



Use Case #3:

Device may experience sudden transmission of non-valid data by a specific sensor or attribute.

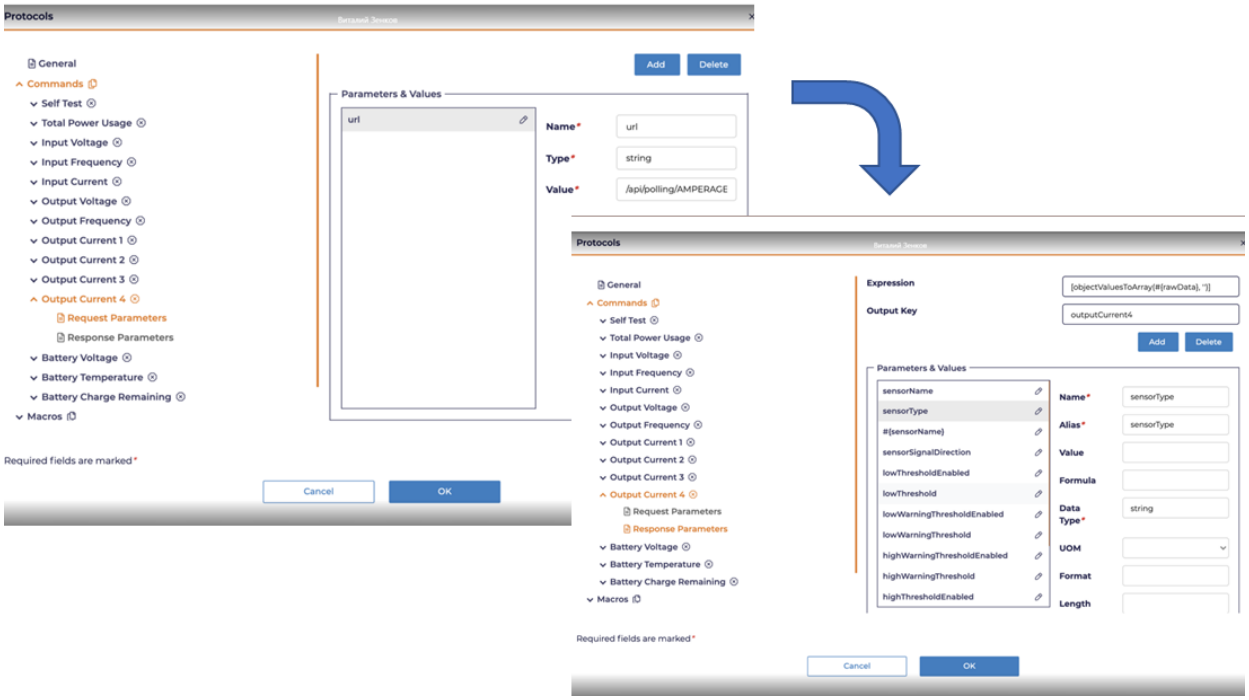
Vixtera's declarative protocol and its macros (executable sequences) allows simple inclusion/exclusion of specific execution commands by using a macro with a set of commands for discovery or polling operations. The commands itself or its specific parameters can be easily removed from execution and operation.



Use Case #4:

Frequent firmware updates

Vixtera's declarative protocol allows easy addition of the commands to map response to the entire model without any change of code. The protocol defines two types of data – facts and sensors. Fact is a key/value information which is not changed frequently. Sensors have a set of standard attributes like type, identifier, name and value. Protocol command defines a processing unit that map the value to some of the system's defined data. So, in case when a hardware vendor wants to add a new feature, the protocol can easily add a new command and map response as an example of a sensor correlated to a certain defined fact.



Use Case #5:

A company X acquired company Y with a multitude of data centers and clouds. The X is in the process of integrating two environments into a single DCIM management structure. A more progressive X uses MQTT protocol to transmit collected data from DCs' devices to the DCIM while, for similar functions, company Y is using an older FTP protocol. Furthermore, thousands of devices in Y's ecosystem are of different brands and types in comparison to devices that are used by X. A recently unified X+Y IT department has got on the effort to integrate all devices into a single ecosystem projected to last for up to 2 years, until Vixtera came to the risqué reducing time to deployment to mere 2 months.

FTP and MQTT are open standard transport protocols that have been developed for the sole purpose of transmitting workloads (data). Both FTP (an older but still widely used) and MQTT (a recently popularized transport protocol) protocols have conceptually similar approach to transport the data – an open standard transport envelope and proprietary topic and payload structure in which collected from devices and/or sensors data transmitted to a centralized server/location. As the result, once data has been collected at the centralized location, it must be structured and then mapped to be understood by any upstream system. So much for a "standard" model.

There're two options to structure collected data, a) write a proprietary software which takes significant time and effort of qualified and trained developers (think about thousand "brown filed" and new devices), or, b) use Vixtera declarative protocol to simply configure data in the payload for a required device type (a single device type may include hundreds of different device models and versions that belong to the same type) - minimum effort is required, and then easily map it to any external system, platform or cloud.

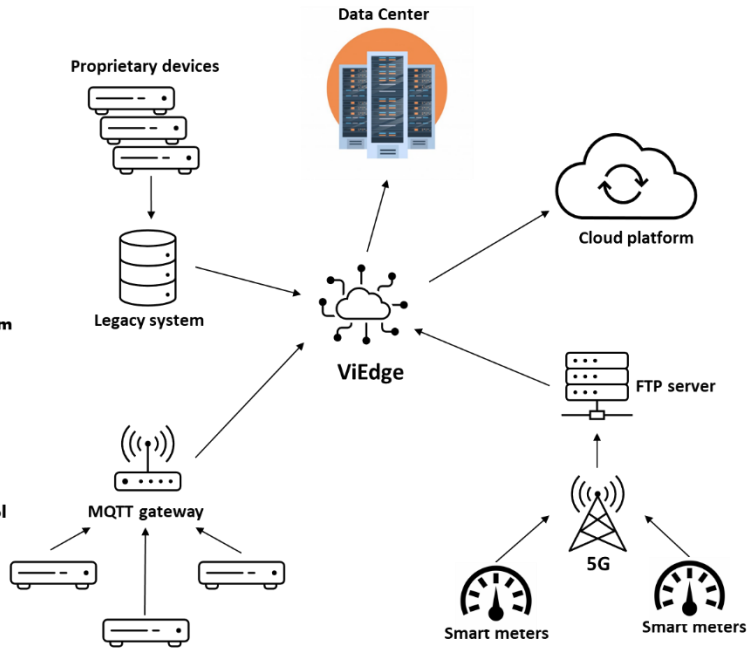
INTEGRATION EFFORT

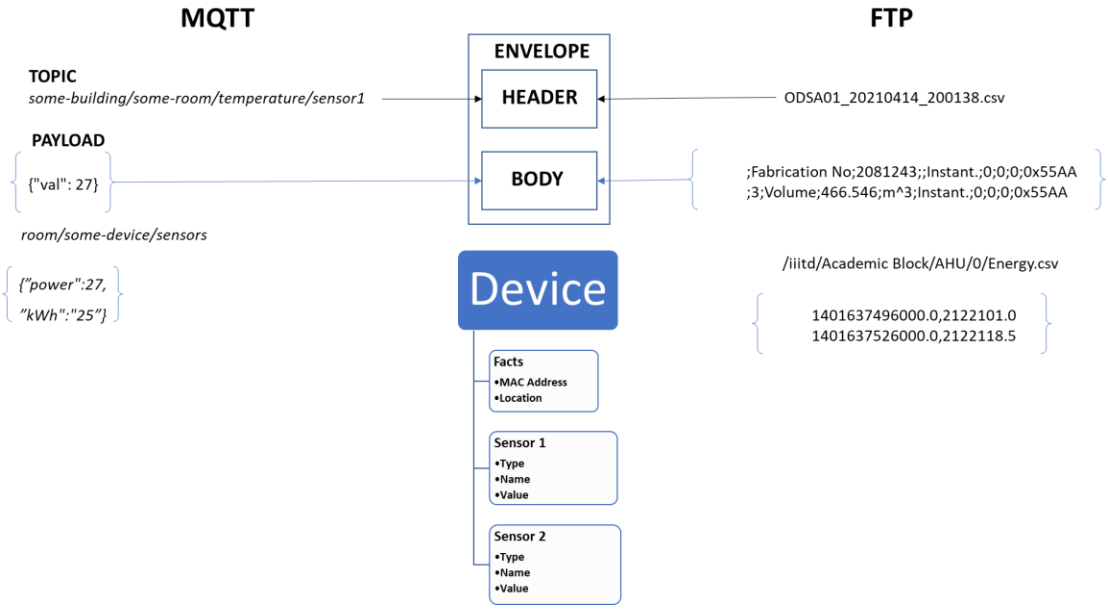
Legacy System Migration Process

1. **Create a software appliance (plugin) to push data into modern system**
2. **Create a software appliance (plugin) to poll data from legacy system (API level)**
3. **Create a software appliance (plugin) to poll data from legacy system (DB level)**
4. **Develop legacy devices support by Modern system**

Vixtera IIoT Edge (ViEdge)

1. **Simple data configuration using Declarative Protocol**





VIXTERA TECHNOLOGY INNOVATION

In summary, Vixtera innovative approach using declarative protocol delivers 100x velocity of device on-boarding, maintenance and management across any transport protocol (same apply to any communication protocol) or device brand. Minimum effort, no drivers, SDKs, APIs or other specialized software is required.