**Multitude of standards**

IoT and M2M technologies are going through a period of explosive growth, which is reflected in the number of new networking and M2M standards rapidly developing to address market needs. Networking standards include LoRa, SIGFOX, INGENU, LTE-NB define, and LPWA technologies for M2M device connectivity, and IoT standards such as AllJoyn/AllSeen, OCF/OIC, Thread, oneM2M and many others provide individual frameworks for device manufacturers.

Multiple IoT standards require similar functionalities to be supported by the IoT devices, and Spirent’s IoT Device Framework can abstract these as Common Services Functions, and they can be classified as per drawing below:

<table>
<thead>
<tr>
<th>Common Services Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Service Layer Mgmt</td>
</tr>
<tr>
<td>Communication Handling</td>
</tr>
<tr>
<td>Data Management and Repository</td>
</tr>
<tr>
<td>Discovery</td>
</tr>
<tr>
<td>Group Management</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Registration</td>
</tr>
<tr>
<td>Security</td>
</tr>
<tr>
<td>Charging and Accounting</td>
</tr>
<tr>
<td>Subscriptions and Notifications</td>
</tr>
<tr>
<td>Network Integration / Device Triggering</td>
</tr>
<tr>
<td>Device Management</td>
</tr>
<tr>
<td>Device Management and Diagnostics</td>
</tr>
</tbody>
</table>

These Common Services Functions, implemented for different standards, mean the same device can be reused in different ecosystem deployments. They also enable creation of the gateway devices with universal, protocol- or standard-agnostic service functionality.

**Challenges facing device and chipset manufacturers**

Creating applications is the driving force behind the IoT, and devices need to enable that functionality. While application enablement is important, there are many different elements device manufacturers need to take into account, and chasing all new versions of the standards, achieving interoperability with other implementations and ensuring API backward compatibility for developers might not be primary focus of the device manufacturers. This is where the Spirent IoT Client Platform makes the difference.
Spirent M2M Core layer supports full protocol abstraction, providing a universal set of APIs for the developers, where the specifics of the particular protocol integration is managed by the respective Protocol Adapter implementation. This approach allows to extend platform to support future IoT and M2M protocols with ease.

Configuration/Provisioning APIs allow applications to set and control all configuration settings and options during run time.

Device Abstraction APIs allow developers to manipulate data models and semantic definitions in accordance with type of the device/solution they are developing.

Common Service Function (CSF) APIs are the primary set of APIs allowing implementation of all relevant common service functions – depending on the type of device (i.e., sensor versus gateway), so the CSF might be not available.

M2M/IoT Protocol Adapters is an abstraction layer which implements support for various IoT/M2M protocols and simplifies adding new protocols on an as-needed basis by the user.

Security Extensions and APIs is an optional module which implements protocol- or /standard-independent security enhancements for end-to-end security.

Analytics Enabler and APIs is an optional module allowing application control data collection and to implement connectivity to the analytics engine of choice.

Testing and Diagnostics Extensions and APIs is an optional module allowing application to support additional run-time/deployment testing and diagnostics functionality, including support for onboarding requirements by the various service providers.

**IoT Device Framework Architecture for constrained devices**

The “things” which comprise IoT can range in its computational capabilities from the super-computers in the cloud to the tiniest, sometimes even barely visible sensors. In many cases those sensors are classified as “constrained devices”, run on the battery and have only a 100K of RAM or so. It is obvious that such constrained devices call for very specific, purpose-built software load which will be capable to run in such a limited environment.

The same Spirent IoT Device Framework is designed to support both full capability and constrained device environments. In the constrained device mode architecture is focused on dramatic reduction in both code size and resource utilization, such as memory, network access and general computation algorithms – but it still offers all the benefits of flexible deployment in various ecosystems.

**Distributed intelligence**

A typical IoT solution might comprise hundreds or thousands of devices communicating to each other and to the service nodes which can be located anywhere in the network, cloud included. All those devices are expected to generate vast arrays of data, some of which can be high priority, and need to be managed, which in some use cases will have to be done within the local network. In the modern computing terms, this is the functionality which is typically classified as “fog computing”.

The Spirent IoT Device Framework enables the “fog computing” paradigm by allowing solution developers to implement data aggregation and service decision logic as part of the extended feature set. Moreover, the role of the devices and nodes can be changed dynamically at run-time to support the most efficient service delivery implementation according to the use case and particular requirements.

**Flexible deployment architecture**

The framework is designed with the ultimate flexibility in mind, allowing various deployment models. Taking advantage of multi-protocol support and design for the vast range of devices from sensors to the gateways, it enables developers to create products for different IoT verticals and ecosystems. The device powered by the Spirent IoT Device Framework can be deployed in any supported ecosystem. Similarly, the gateway built on the Spirent Device Framework will be able to support and manage a network of disparate sensors and devices even when non-interoperable protocols are utilized.

**Advantages of the Spirent IoT framework solution**

- Well defined APIs and data abstractions
- Multiple API classes
- Additional low-level API layer for maximum flexibility
- Enhanced end to end security
- Available analytics integration
- High degree of customization
  - code size
  - conditional compilation – include only what is necessary
  - memory consumption
  - performance optimization
- Optimization for low CPU power environments
- Future proof – we track and implement all new versions of the standards
- Backward-compatible APIs
- Support for various transports (HTTP, MQTT, CoAP, WebSockets)
- C APIs
- Full source code
- Multiple OS supported
- Highly portable
- Hardware and OS abstraction
- Protocol Abstraction
- Support for testing and diagnostics, including service provider onboarding