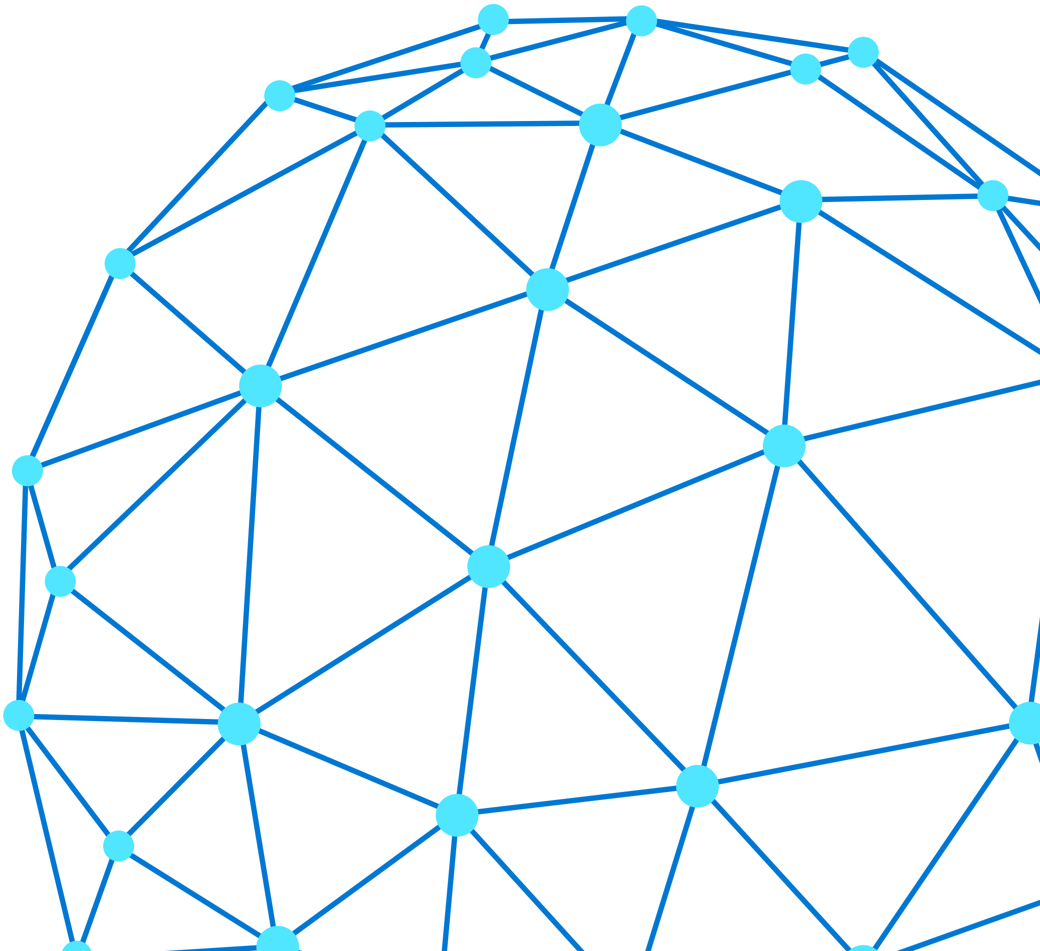
Microsoft Azure

Cellular connectivity options immediately available to users of Azure Sphere

Nicholas Chen
Microsoft



July 2020

Contents

Executive summary

Microsoft Azure Sphere Operating System (OS) currently supports two kinds of network connectivity: Ethernet (via an external Microchip ENC28J60 Ethernet chip connected over the SPI bus) and Wi-Fi. For customers that require connectivity to the Internet via cellular networks, one option is to use external hardware to route the Ethernet or Wi-Fi to the cellular network (Figure 1). A Wi-Fi cellular "hotspot" is an example of one such device. Since security-critical functions like certificate-based authentication and software updates can only be performed through OS-supported network connections, other kinds of connections to external cellular hardware should be avoided.

This router-based architecture results in some implications that customers should be aware of. First, it is critical that customers understand that the security-related functionality and guarantees that Azure Sphere offers do not extend to the router and cellular connectivity portions of a cellular-router-based system. Securing these parts of the system are the responsibility of the entities deploying and administering the system. Since the overall security of a system is governed by the weakest link, customers must ensure that they are able to secure and maintain these parts of the system or find trusted partners who can do it on their behalf. Also, in this architecture, Azure Sphere OS is unaware that the connection to the Internet is over a cellular link. As such, the OS cannot mitigate issues arising from cellular network performance nor will the OS alter its operation to be more optimized for a cellular link.

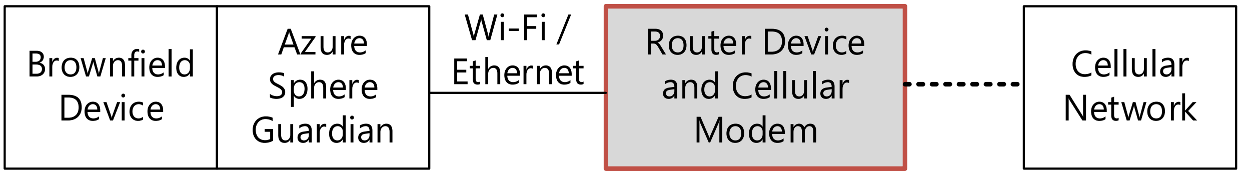


Figure 1. High-level architecture of using a cellular-enabled router to enable cellular connectivity. Azure Sphere does not provide security to the router and cellular modem parts of the implementation (outlined in red).

Cellular connectivity

Today, Azure Sphere OS supports LAN interfaces in the form of Wi-Fi and Ethernet. To reach the WAN (i.e. the Internet), Azure Sphere depends on a router device. Azure Sphere OS is oblivious to and agnostic about the form the WAN connection takes. For instance, Azure Sphere OS does not care whether connectivity to the Internet is over a company's fiber optic link or a residential DSL line. Cellular connectivity would be no different: a router that connects to the internet via the cellular network would look identical to Azure Sphere OS and will work without requiring any new OS functionality. As such, cellular connectivity based on connecting over Wi-Fi or Ethernet to a separate router device is an immediately realizable option for customers using the Azure Sphere MediaTek MT3620.

The first decision customers that wish to connect Azure Sphere devices to the cellular network must make is whether to use the MT3620 on-chip Wi-Fi or external Ethernet connectivity. Key advantages of each choice are presented below.

Wi-Fi	Ethernet
Key advantages • Does not require additional chips on Azure Sphere device side • Availability of many off-the-shelf cellular "hotspot" devices • No cables, simplifies installation • Ability to share a single access point across multiple devices • Preserves the wired Ethernet link for private connection scenarios	**Key advantages** • Less susceptible to interference • Less configuration required (on both router end and on Azure Sphere device end) • Possibly fewer certification requirements (if building custom hardware) since radio is unused • Wired connection between Azure Sphere device and router removes wireless attack vector

Having decided on the interface, it is then a matter of sourcing or building a network router that will forward traffic to and from the cellular network. In selecting a router, a decision needs to be made between using an off-the-shelf cellular router product or developing custom hardware and software. The advantage of custom hardware is higher integration, which can reduce cabling and enables a more compact, monolithic device. The potential downsides of custom hardware are a longer development time due to hardware testing and verification as well as a possible need to undergo regulatory and operator certification.

The architecture for a custom solution that is integrated onto a single board is shown in Figure 2. Note that this design does not fundamentally differ from the basic router-based architecture using off-the-shelf hardware discussed above, it simply combines the different pieces onto the same circuit board.

In this design, the router chip can be any MCU or MPU with IP routing capability. Depending on chip selection, a second SPI Ethernet chip or an Ethernet PHY may be needed for the router chip to send and receive Ethernet traffic. Using this architecture, implementers are free to select the cellular modem that is most appropriate for a given application. The requirement when selecting a cellular modem is that the modem must be able to forward IP packets directly onto the cellular network. Using cellular modems that are pre-certified by mobile operators will minimize certification time for the overall system. Finally, the router chip that the cellular modem is attached to can optionally expose information from the cellular modem back to customer applications through a separate control channel.

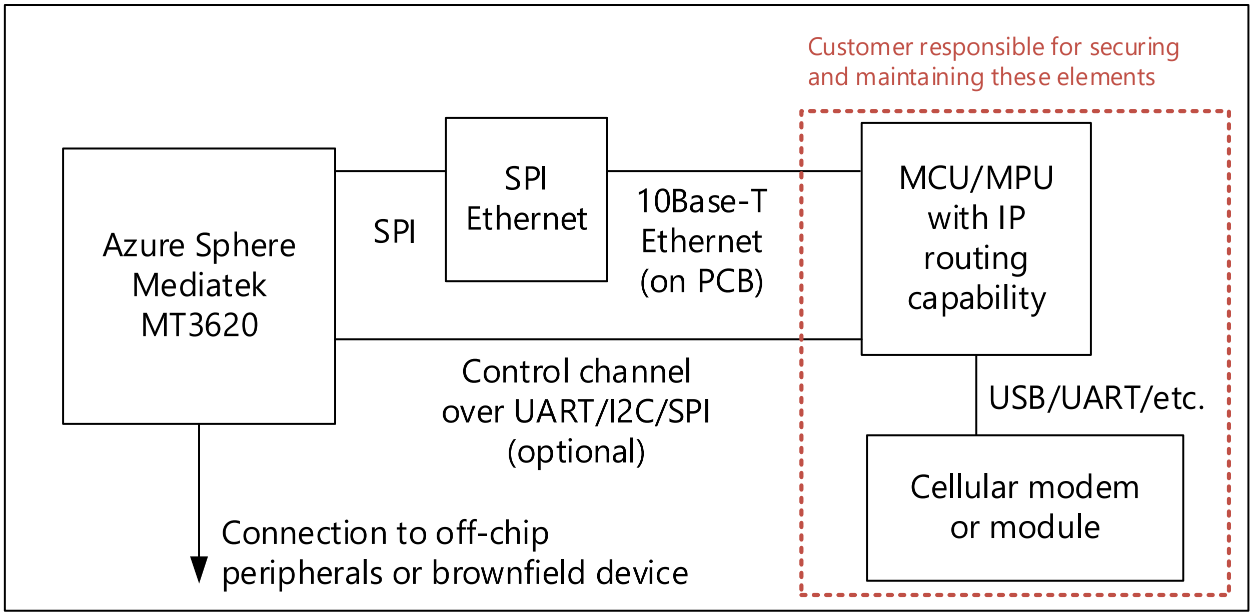

Customer responsible for securing and maintaining these elements
Azure Sphere Mediatek MT3620
SPI
SPI Ethernet
10Base-T Ethernet (on PCB)
MCU/MPU with IP routing capability
USB/UART/etc.
Cellular modem or module
Control channel over UART/I2C/SPI (optional)
Connection to off-chip peripherals or brownfield device

Figure 2. An architecture for enabling cellular connectivity on the MT3620 that integrates the router chip on the same circuit board as the MT3620. The red box denotes the portion of the system that customers or customer's partners are responsible for securing, maintaining, and managing.

Factors to consider when evaluating cellular solutions

To help customers evaluate whether pairing Azure Sphere with a router is an appropriate solution for cellular connectivity in their application, we wish to identify several factors that customers should be aware of when creating a system using this architecture.

Security considerations

The inclusion of an external router in the solution described above, which is essentially an embedded computer connected to the Internet, introduces security risk. Connecting to the Internet through a cellular connection introduces similar network security risks that would be present when connecting through other routing devices (such as a commercial Wi-Fi access point/router connected to the Internet). Critically, entities responsible for device and network security must be aware that Azure Sphere security properties extend up to the Ethernet or Wi-Fi interface exiting the MT3620, but do not extend to the cellular router.

In all implementations, entities must ensure that the non-Azure Sphere parts of the system are properly secured (i.e., defense in depth, appropriate access control mechanisms, and software regularly updated in response to new vulnerabilities). Failure to secure the router portion of the system may result in the Azure Sphere device losing connectivity to the Internet or may lead to more serious problems should the router portion become compromised. Customer or their partners implementing the solution should acknowledge and accept the incremental risk this solution represents, recognizing that a failure on the router side may require physical servicing or device replacement.

Particular care should be taken with designs in which the MT3620 and router parts are combined into a single device since the boundary between these elements may not be obvious. If the implementation uses a secondary control channel between the MT3620 and the cellular router chip, the customer app receiving the input should validate the messages arriving from the router chip. This precaution needs to be taken because if the router chip is compromised, the messages sent from the router chip over the control channel may be malformed or malicious.

Cost considerations

The incremental additional cost compared to a non-cellular Azure Sphere solution depends heavily on the hardware choices used to implement the system. Using commercial, off-the-shelf, hardware will generally result in higher bill of materials cost versus custom, cost-optimized hardware created using low-level electronic components. However, custom hardware designs typically introduce additional development costs that include creating and testing the software for the router and the possibility of having to undergo regulatory or mobile operator certification on account of the hardware being new and not having undergone prior testing. The cost of operating the device, which includes the cost of sending and receiving data, is another factor to consider. The data usage for Azure Sphere servicing and updates, which is on top of customer data transfer needs, is estimated to be tens of megabytes per month. The choice of operator, network, and service plan must take these data transfer requirements into account.

Functionality considerations

Lastly, a limitation of the router-based architecture is that the Azure Sphere OS has no awareness of the existence of the cellular connectivity. In some cases, hiding these details may be a useful simplification for the application running on the Azure Sphere device. However, the architecture prevents the OS or applications from being aware of and taking action in situations that may arise in cellular applications. For example, cellular network performance (e.g. latency and throughput) can vary dramatically—these situations may prevent the device from authenticating with the Azure Sphere Security Service and receiving updates. Customers must make sure that these situations are accounted for. Finally, many off-the-shelf cellular routers do not expose additional functionality that cellular networks provide, such as SMS or Cellular Assisted GPS positioning. If these features are needed, then a custom hardware solution may be required.

Conclusion

Connectivity over cellular is possible today with the Azure Sphere MediaTek MT3620. The mechanism for achieving this is through the use of a cellular router device or SoC that links the Wi-Fi or Ethernet interfaces exiting the MT3620 with a cellular connection to the Internet. Customers and partners that are interested in creating devices based on this architecture should be aware of the security implications, costs, and functional limitations of such a solution. During the current timeframe, when Azure Sphere OS only supports Wi-Fi and Ethernet connectivity, other cellular connectivity architectures that differ from the router-based approach described in this document will result in a device that cannot make use of critical Azure Sphere functionality, such as software updates, and should not be employed.

Appendix:
Cellular connectivity architectures that should not be used

Connecting to a cellular modem using an MT36230 UART (ISU)

Many cellular modems designed for IoT applications are accessed over a UART interface. Since the MT3620 provides several customer-accessible UART interfaces (i.e., ISU0-ISU4), designers may be tempted to connect the cellular modem to one of these interfaces. In so doing, the customer app can gain access to the cellular network via the modem and can send and receive data. Unfortunately, since Azure Sphere OS is unaware of this interface, communications with the Azure Sphere Security Service, such as device authentication and attestation (DAA) and software updates, cannot occur.

Using an ISU to send data to an external MCU that has cellular connectivity

Another proposed configuration is to use one of the customer-accessible communication interfaces (e.g. I2C, SPI, UART) to send device data to a second MCU, which can then connect to the Internet via a cellular modem. Although the use of a second MCU may make this solution appear similar to the router-based architecture discussed in the main document, the critical difference is that the OS is unaware of the connection to the Internet since it is not over Wi-Fi or Ethernet. This configuration resembles the cellular modem over UART design above, in which the communication interface is only visible to the customer app. As a result, updates and DAA cannot occur over this channel.

Emulating/faking the SPI Ethernet chip

One proposal we have encountered is to program a second MCU to emulate the behavior of the SPI Ethernet chip that the Azure Sphere OS supports. The second MCU would play the role of a router and forward the traffic to and from the MT3620 via a cellular modem. In this configuration, the Azure Sphere OS believes it is talking to the supported SPI Ethernet chip and communications with the Azure Sphere Security Service may continue to work.

We discourage this approach because of the potential to introduce stability issues into the Azure Sphere OS if the behavior of the emulation differs from the actual chip. The SPI Ethernet drivers in Azure Sphere were not designed to cope with this kind of unexpected behavior on the part of the SPI Ethernet chip. Consequently, taking this approach can result in the loss of network connectivity, incorrect system behavior, or system restarts.

Microsoft Azure