**20. Cloud Native Trends**

Shivakumar R Goniwada[**1**](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_20_Chapter.xhtml#Aff2)

(1)

Bangalore, India

I hope by now everyone has a clear view of the end-to-end lifecycle of cloud native systems from architecture to design to development to operations.

What’s next? How will the cloud native journey progress? To provide the answer to this question, I will cover cloud native future trends in this chapter.

The pace of technological change has accelerated. Customer demands are more pronounced, and competitive threats have grown more unpredictable. Change is happening quickly, and industries are racing for their position in the world. COVID-19 impacted the world in ways no one could have predicted, and organizations are adopting IT in new ways.

Gartner, Forester, and other researchers publish trend reports every year about the future, and I have been following all the research for quite some time. I’ll share my thoughts of where things will be going from now.

ThoughtWorks publishes a technology radar report on various techniques, tools, platforms, languages, and frameworks. They update the technologies regularly based on maturity and industry trends.

I will divide the future trends into these two themes:

* Technology trends for cloud native
* Technology trends across industries

There is a lot more you can find via Cloud Native Foundation, Technology Radar, and other leading consulting firms.

* Cloud native journeys related to Kubernetes on the edge, low-code platforms, GitOps, etc.
* Industry trends like 5G, quantum computing, digital twin, etc.

**Cloud Native Trends**

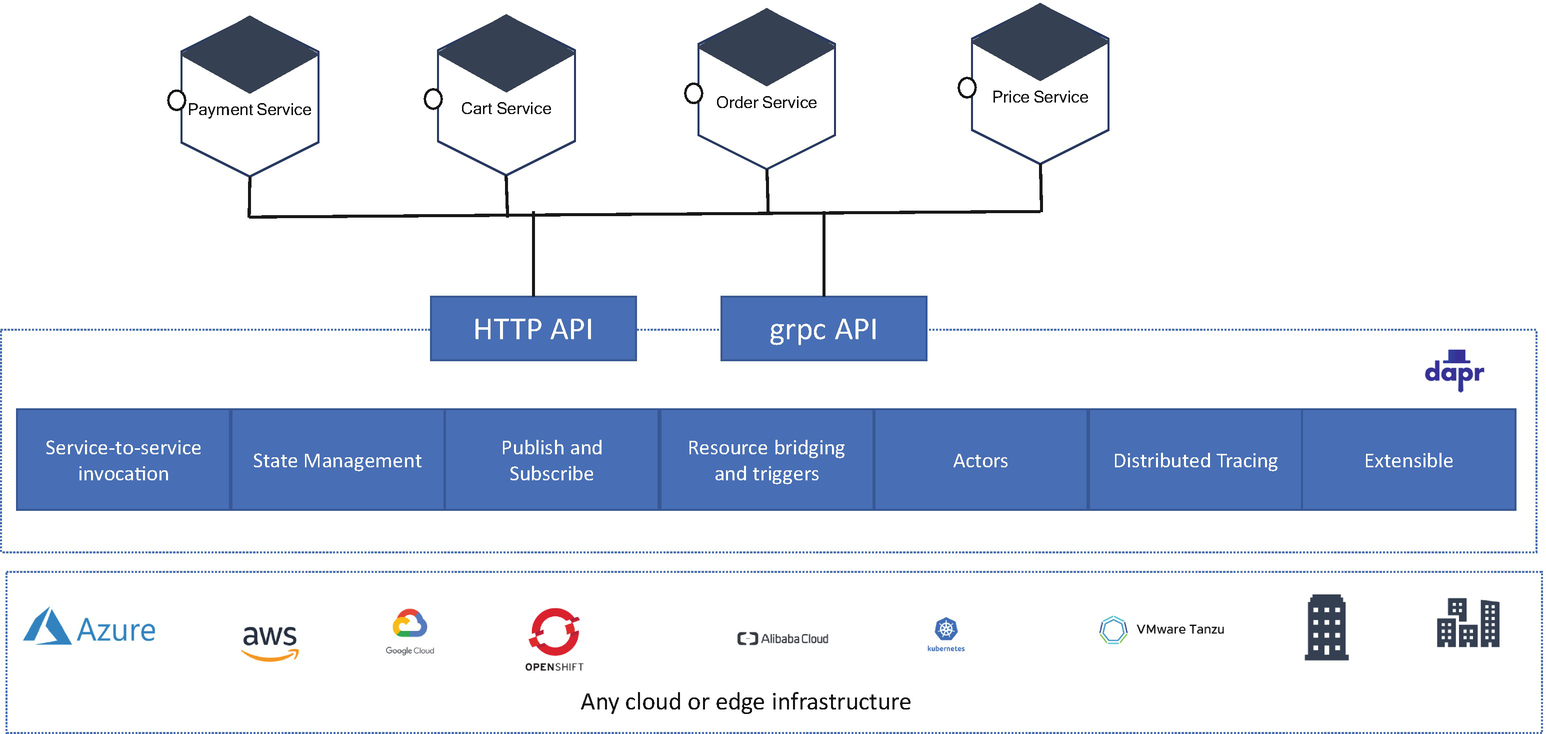
In this section, I will provide brief details of a few trends that are related to cloud native architecture and design.

**Designing for “-ilities”**

In a cloud native world, architecture is reprioritizing the “-ilities” such as resiliency, observability, portability, etc. These cloud native services are distributed over the wire. If something goes wrong with the services, you not only can observe them but also can fix them. You had to build all these things yourself, and it was very complicated, but as we progress, there are now more technologies available to handle them. For example, service meshes and event meshes take care of some of the “-ilities.”

**Cloud Native Architecture**

Event-driven architecture has existed for a while, but based on my experience, engineers still need to master it when writing code. Various software is available to help you to build event-driven systems. For example, the Distributed Application Runtime (Dapr) helps you to build event-driven resilient distributed applications whether it b in the cloud or hybrid or on-premises, as illustrated in Figure [20-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_20_Chapter.xhtml#Fig1). Dapr codifies the best practices for building cloud native services into open, independent building blocks that enable you to build the business logic in your choice of language.



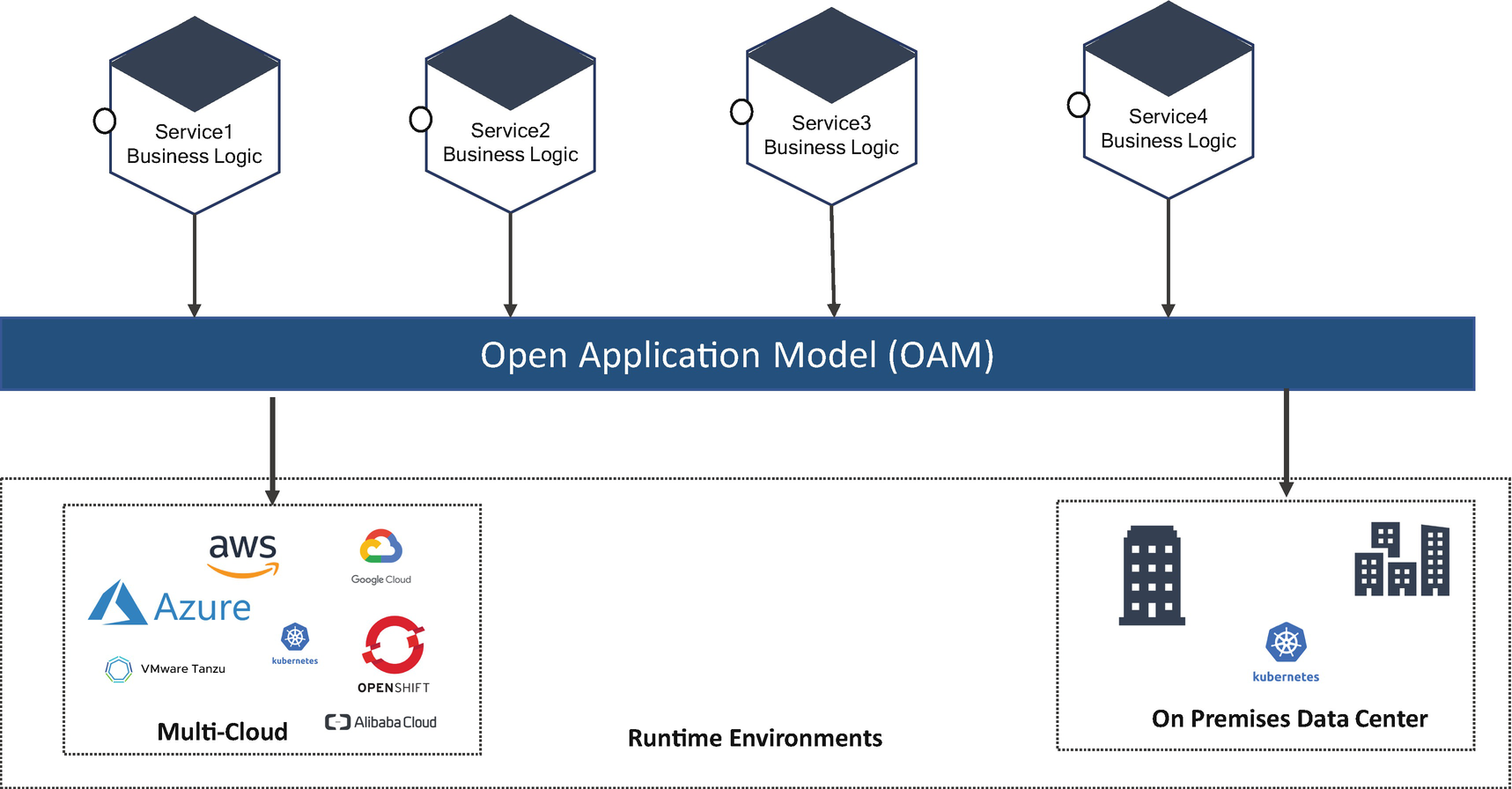
***Figure 20-1***

Dapr architecture

Dapr reduces the burden on engineers to implement all the cloud native building blocks such as state management, publish/subscribe, observability, and secrets. You can find more details at <https://dapr.io/>.

**Open Application Model Specification**

As shown in Figure [20-2](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_20_Chapter.xhtml#Fig2), the Open Application Model (OAM) specification is a runtime-agnostic specification for defining cloud native applications. This specification helps you to focus on application business logic rather than any container, orchestration, or infrastructure-related tasks. This specification brings modular, extensible, and portable designs for modeling application deployment with a consistent higher-level API.



***Figure 20-2***

OAM

As cloud native is maturing, there is a need to have a well-defined and coherent model that represents the complete application, not just a template but a clear specification. The Open Web Foundation, a consortium of many companies, created an OAM specification for describing services so that the description of the services is separated from the details of how the services are deployed onto and managed by the infrastructure. You can find more details at <https://oam.dev/>.

**Web Assembly**

There is growing demand from customers in terms of performance, highly optimized communication, and experience; you can’t just depend on JavaScript anymore. Web browsers are capable of rendering user interface code, and modern architecture requires a heavy-duty task to communicate effectively. Web Assembly helps you to do this. The specifications are designed to do compilation within a browser for other machine languages such as C, C++, etc.

Web Assembly (WASM) is a new type of code that can be run in web browsers and provides high performance. It is not for engineers to write code, but it is designed to be an effective compilation target. You don’t need to know how to create Web Assembly code because it can be imported into a web application. It was created as an open standard under the W3C Web Assembly community group. The following are the use cases for WASM:

* Game development
* AR/VR live application
* Video editing
* Image recognition

You can find more details of WASM at <https://webassembly.org/>.

**Data Gateways**

In the cloud native age, you already are familiar with API gateway and microservice architecture. The API gateways streamline your APIs to the external and internal worlds from microservices. Microservice principles provide you with the ability to embrace polyglot persistence; such polyglot persistence requires an API gateway type model for data. Like an API gateway, a data gateway offers abstraction, security, scaling, federation, and contract-driven development, etc.

As part of the polyglot persistence, each microservice can have its own storage. Some services require relational databases, and others require NoSQL, graph databases, caching, etc. In modern technology, just using data for microservices is not enough; you need data to be exposed to data analytics platforms. This is where the data gateways help; they are similar to API gateways. API gateways work with the network, but data gateways work with data. The following are a few features of a data gateway:

* It abstracts away the physical data store and its specifics. This gives you to the freedom to alter, migrate, and decommission databases.
* It understands the different data models and applies role-based access management with a fine-grained security model.
* It speeds up access to all kinds of data sources by caching data and providing materialized views; it can understand the queries and optimize them based on the capabilities of the data source.
* It can act as a data federation layer.
* It allows a schema-first service like contract first for API gateways.

The data gateway tools are Apache Drill, which is a schema-free SQL language for NoSQL database; Teiid, which is a data federation engine; PrestoDB, which is a distributed SQL query engine; and AWS Athena, which is an ANSI SQL-based interactive query service for analyzing data tightly with S3.

**HTTP/3**

This is a third and major version of HTTP used to exchange information over the Web. It provides the same features as HTTP/2: request methods, status codes, and message fields. HTTP/2 uses TCP as a transport, but HTTP/3 uses QUIC as a transport layer. The QUIC (it is not an acronym) is Google’s transport layer protocol, and later the Internet Engineering Task Force (IETF) adopted it. The QUIC is a reliable and secure transport protocol and addresses the shortcomings of HTTP/2 over TCP and TLS. QUIC is a key element of HTTP/3 built on top of UDP and attempts to solve the major issues experienced when using TCP-like connection establishment latency, multistream handling, etc. Google uses QUIC for its server’s traffic. As of this writing, it is still in draft form, but already 71 percent of running browsers support this. You can find more details at <https://quicwg.org/base-drafts/draft-ietf-quic-http.html>.

**RSocket and Reactive Streams**

For cloud native services, HTTP is the de facto standard for communication; it is very well suited for services but may not be suitable for all use cases. If you want to communicate other than request-response, then it is difficult. You can achieve it, but it is not what the protocol was developed for.

RSocket is a new messaging protocol, and it is designed to solve some common cloud native services communication drawbacks. RSocket is a flexible protocol that works with TCP or WebSockets. You can do binary messages without any conversion and with control of multiplexing, back-pressure, resumption, and routing, and you can use it for fire-and-forget, streaming, and also request-response. This is best suited for reactive architecture and ideal for high-performance and high throughput services. There are many companies like Netflix, Alibaba, Facebook, etc., that have adopted this protocol for respective use cases.

**Low Code/No Code**

Development platforms with visual software development environments allow enterprise engineers to drag and drop application components, connect them, and create web or mobile apps with minimal hand-coding. They help to build quickly instead of writing line-by-line code. Various companies are coming up with low-code tools like AWS HoneyCode, Pega, etc.

These tools help to build custom software using the following steps:

1. 1.

Organize data in tables.

1. 2.

Build apps with visual tools in a drag-and-drop approach.

1. 3.

Use automation to replace manual steps.

All the cloud vendors started with low-code services to build an app. For example, AWS HoneyCode helps you to build software by using existing services like Lambda, S3 buckets, etc., and HoneyCode generates relevant source code and API code for applications. You can find more details in the respective tool providers. These tools have started emerging due to the skill gap in cloud native services.

**Actor Model**

The actor model is a design pattern that allows your team to focus on an application’s business logic rather than low-level protocols. Self-healing, lightweight, and event-driven, actors take a drastically different approach to messaging and processing. This helps you to build distributed and reactive systems.

The actor model is not new; in 1970 Carl Hewitt and Alan Kay were running into memory issues and slow programs. Their intention at the time was to create message-passing systems. The actor is a computer process or function; you are passing some messages to the actor by calling functions, and it returns some messages.

Building reliable, scalable, event-driven, and distributed services in a multicloud environment is not simple. Kubernetes and containers are helping us to meet the previous characteristics. To take advantage of these technologies to build your system, you use an actor model for simplification. You can relate the actor model to microservices; in this way, the actor can be microservice clients, event publishers, event handlers, message brokers, distributed loggers, error handlers, observables, etc. In a nutshell, everything around your cloud native services is an actor. You can find more details on InfoQ webinars.

**Kubernetes on the Edge**

The edge is a topology that brings computation and storage to the location where it is needed to improve response time and save bandwidth. Kubernetes provides a complete edge computing solution with separated cloud and edge modules. The control plane of Kubernetes resides in the cloud with scalability and extensibility, and at the same time the edge can work in offline mode. These can be done by using the KubeEdge software; it is lightweight and containerized and supports heterogenous hardware at the edge.

KubeEdge is built on Kubernetes and provides core infrastructure support for networking, services deployment, and metadata synchronization between the cloud and the edge. With the core business logic running at the edge, much larger volumes of data can be secured and processed locally where data is produced. This is one of the trends and works in parallel with edge computing. You can find more details at [https://kubeedge.io/en/#home\_slider](https://kubeedge.io/en/%2523home_slider).

**GitOps**

This is a way of implementing continuous deployment for cloud native services. It is an operational framework, and it requires you to describe observability for systems with declarative specifications that adhere to continuous everything principles. It focuses on operating infrastructure and operations capability with DevSecOps and Kubernetes and by using infrastructure as a code.

The following are the main principles of GitsOps:

* The entire system is described declaratively.
* Everything is versioned in the Git repository.
* Approved changes are automatically applied to your system by using declarative configuration.
* Software agents ensure the correctness and alert on divergence on expectation and state.

GitOps is part of infrastructure as code; it checks the status of the infrastructure automatically and changes according to that. You can find more details in Chapter [17](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_17_Chapter.xhtml).

**General Trends Across Industry**

This section covers some trends.

**5G**

5G is the fifth-generation technology standard for broadband cellular networks. It is a new global wireless standard after 1G, 2G, 3G, and 4G networks. 5G enables a new kind of network that is designed to connect virtually everyone and everything including machines, objects, and devices.

5G wireless technology is meant to deliver higher multigigabyte per second peak data speed, ultra-low latency, more reliability, massive network capacity, increased high availability, and a more uniform user experience.

According to the various experts quoted in a recent economic study, 5G is driving a global growth of $13.2 trillion, 22 million new jobs, and $2.1 trillion in GDP growth.

The scope of 5G will ultimately range from mobile broadband services to next-generation architecture in automobiles, financial, manufacturing, consumer products, connected devices, etc.

**5G Technology**

The initial 5G New Radio (NR) like LTE in the 4G specification was completed in June 2018. 5G NR is a new Radio Access Technology (RAT) developed by the Third-Generation Partnership Project (3GPP) for the 5G mobile network.

3GPP is an umbrella group of several standards organizations that develop protocols for mobile telecommunications.

3GPP maintains the following standards:

* GSM and the related 2G and 2.5G standards
* UMTS and the related 3G standards
* LTE and the related 4G standards
* 5G NR and the related standards

Long-Term Evolution (LTE) was developed as a 4G standard; this was a global standard for wireless technologies and is presently used.

**5G Features**

5G comes with various features and capabilities like network slicing, orthogonal frequency-division multiplexing (OFDM), and multiple input and multiple output (MIMO).

5G NR uses two frequency ranges and operates on a new frequency spectrum: millimeter-wave (MM wave).

* Frequency range 1 (FR1): Sub-6GHz frequency bands
* Frequency range 2 (FR2): 24.25 GHz to 52.6 GHz

The two main major trends behind 5G are as follows:

* Digital technologies with more mobile access across the globe that can carry ultra-definition video, data, and Services in Hand (SIH) across industries to the user
* Internet of Things (IoT), where large numbers of smart devices communicate over the Internet with ultra-high-speed

**Advantages of 5G**

It is substantially different to define an architecture for 5G compared to earlier networks. The following numbers are based on various industry leaders’ testing, and these numbers are illustrative:

* Latency: Less than 1 ms
* Latency end to end (device to the core): Less than 10 ms
* High download speeds: 10 Gbps
* Base stations: Small cells
* OFDM encoding: 100 to 800 MHz channels
* Connection density: 100 times greater than LTE
* Energy efficiency: Greater than 90 percent improvement over LTE
* 1,000,000 IoT devices per square kilometer

5G speeds will be enabled by massive MIMO communication in the millimeter-wave frequency range. With this standard, 5G is able to provide significantly higher mobile broadband throughput with its enhanced mobile broadband mode.

**Cloud Native and 5G: Network Slicing**

For your project, there are many business scenarios where you might need to build a dedicated network to serve the customer for each business scenario. In this case, you need to have a separate network and separate management, and there is no single platform to host all dedicated networks on one, i.e., like virtualization.

*Network slicing* is a method of creating multiple unique logical and virtualized networks over common multiple networks. It is the embodiment of the concept of running multiple logical networks as a virtual independent network. Network slicing is the ability to customize the capabilities and functionalities of the business use cases.

With network slicing, each slice can have its architecture, management, and security to support specific use cases. While resources are shared across network slices, the capabilities of the network such as capacity, connectivity, reliability, and latency can be customized in each slice.

Automation is the key component in creating and building network slicing. As you might need to build hundreds of network slices, you need to create similar ones like infrastructure as code or network as code. You can find more details on Qualcomm.com and Ericsson.com.

**Digital Twin**

The manufacturing industry has been facing a lot of challenges such as efficiency, resiliency, throughput, quality, automation etc. but research is underway to adopt modern techniques and some global manufacturing organizations are doing research and proof of concepts on digitizing the process and machine behaviors in shop floors.

The digital twin platform is an effective means to reflect the physical status in the virtual space. It breaks the barriers between the physical world and the digital world of manufacturing.

*“Digital Twin is a sensor-enabled digital model of a physical object that simulates the object in a live setting.”*

*—Dr. Michael Grieves*

A digital twin is a digital representation of the physical world. The technical capabilities behind digital twins have expanded to include buildings, industries, people, processes, households, etc.

A digital twin is essentially a computerized mirror of a physical asset and/or process, in other words, a virtual replica that relies on real-time data to mimic any changes that occur throughout the lifecycle.

The digital twin idea was first conceived by Michael Grieves at the University of Michigan in 2002. The right technology at the time was unavailable, but now it is the right time to consider the evolution of technologies such as AI, ML, IoT, cloud, and quantum computing.

A digital twin is a vital software tool to help engineers to understand not only how products are performing but how they will perform in the future. Analysis of the data from the connected sensors, combined with another source of information, allows us to make these predictions.

**Why a Digital Twin?**

A digital twin is a simulation model that represents a machine or a business process.

The digital twin will help manufacturing and business: the behavior of machines with predictive and preventive analysis. It improves process and functioning, reduces industrial accidents, etc. The following are the use of the digital twin:

* Optimize asset behavior by applying real-time analysis to the virtual object and modifying the behavior of the real object system.
* Suggest the optimization to the real object system.
* Observe the current real object system behavior and status by applying sensor readings to the virtual object and observing its behavior.
* Observe the historical behavior and the status of the asset.
* Simulate the real object system, which helps to optimize the configuration.
* Predict the future behavior by running predictive and behavioral analysis.

**Digital Twin Implementation**

Digital twins can be implemented in multiple ways depending on the type of industrial machine that you want to create the twin.

* *Digital twin prototype (DTP)*: The digital twin prototype describes the physical artifact. It contains the informational sets and virtual versions of real objects. This provides information such as the 3D model, bill of material (BOM) with detailed specification (BOS), bill of processes (BOP), bill of services, etc.
* *Digital twin instance (DTI)*: A digital twin instance is a digital twin always linked to the real system throughout the life of that physical product; it contains an exact 3D model, BOM, BOP, BOS, etc., along with the results of any measurements and tests on the instances and a service record with past services and replaced components. Operational states are captured from the actual sensor data in real machines.
* *Digital twin aggregate (DTA)*: This is the aggregation of all the DTIs and captures the group of data structures from the DTI. It queries all the data in DTIs and analyzes them together. It continually examines sensor readings and correlates those sensor readings.
* *Digital twin environment (DTE)*: This is the end-to-end environment setup to operate digital twins. The operations include receiving data from real machines, analyzing the data, doing predictive analysis, performing behavioral analysis, etc.

You can find more details at Engineering.com.

**Quantum Computing**

Quantum computing is a computing system based around quantum theory. Quantum theory is the theoretical basis of modern physics that explains the nature and behavior of matter and energy on the atomic and subatomic levels. Quantum computing uses a combination of bits to perform computational tasks and to perform a calculation based on the probability of an object’s state before it is measured. It does not use normal computer 0s and 1s as conventional digital computers do, but it uses quantum bits or qubits to encode information as 0s, 1s, or both at the same time.

**Why Quantum Computing?**

Existing servers like virtual machines are flexible and offer higher computational performance when solving specific problems. These machines are increasingly used to solve a certain variety of use cases, and they outperform CPUs, from low-latency stock trade validation to streaming data to computationally intensive workloads.

* *General-purpose central processing units (CPUs)*: Today’s CPUs execute programs by performing a long sequence of basic arithmetical, logical, control, and input/output operations.
* *General-purpose* *graphics processing units (GPUs)* : It is an accelerator designed for parallel calculations. GPUs outperform CPUs when large blocks of data are processed in parallel such as graphics processing, AR/VR, training ML models, etc. The drawback is that the complex operations cannot be broken down into independent straightforward calculations.
* *A* *field programming gate array (FPGA)* : This is an accelerator that is like an ASIC but can be configured or reprogrammed after manufacturing. It is more efficient and powerful than GPUs and CPUs. FPGA is good for parallel applications and DNA sequencing, etc.
* Application-specific integrated circuity (ASIC) : This is an accelerator hardwired and used in Google’s Tensor Processing Units (TPUs). It is very fast and power-efficient, and the logic is written in the hardware itself. It does not require any translators or execution area to execute the software code.

The previous processing units can solve specific use cases, but you need a paradigm shift to process large computational processes; therefore, you need to shift from bit to qubit.

As you know, quantum is based on the principles of quantum mechanics and increasingly attracts the interest of automotive, retail, and distribution networks because of its ability to efficiently solve complex problems.

**Potential Use Cases**

Here are some use cases:

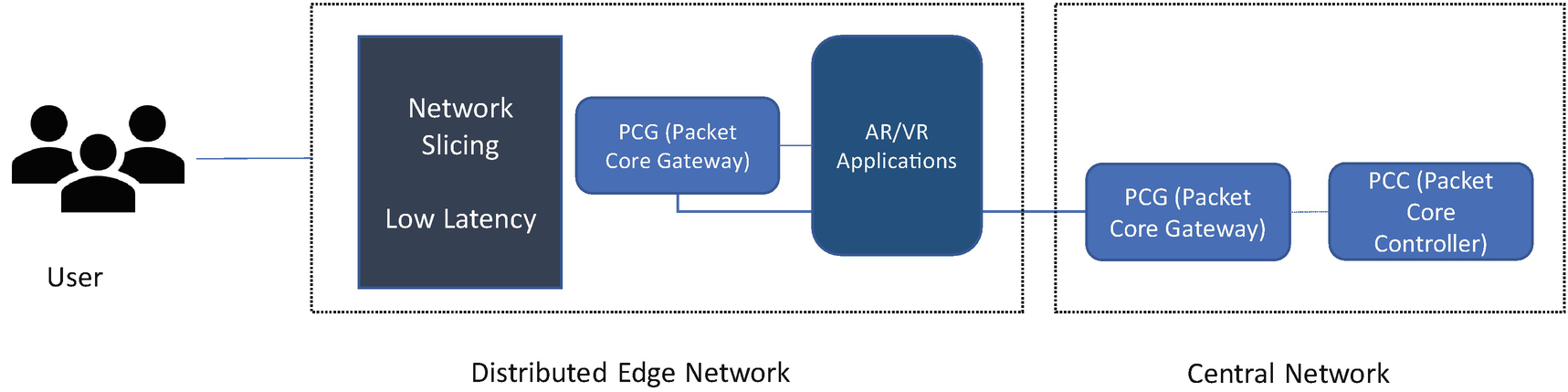
* *Genome sequencing* : A large amount of genomic patient data and genome-wide association studies are researching for cross-referencing genes and diseases, which requires enormous computational efforts.
* *Radiation therapy* : Having a radiation plan can minimize damage to the surrounding healthy tissues and body parts. Arriving at the optimal radiation requires many simulations until an optimal solution is achieved.
* *Transaction security* : Connected devices make use of secure encryption for transactions in online retail stores and credit card payments.
* *Molecular modeling* : Quantum helps explore the properties of new materials and identify the characteristics in the chemical structures of useful materials.

There are many more use cases where you may require quantum computation. You can find more details on quantum computation from IBM, Google, Azure, or AWS.

**Extended Reality**

Extended reality (XR) provides a form of digital sensory awareness that is driven by the physical world around us. It delivers real-time, highly personalized, and contextual experiences using a combination of audio, visual, and even tactile devices. It is an umbrella term for all immersive technologies such as augmented reality (AR), virtual reality (VR), and mixed reality (MR). All immersive technologies extend reality either by combining the virtual and real worlds or by creating a fully immersive experience.

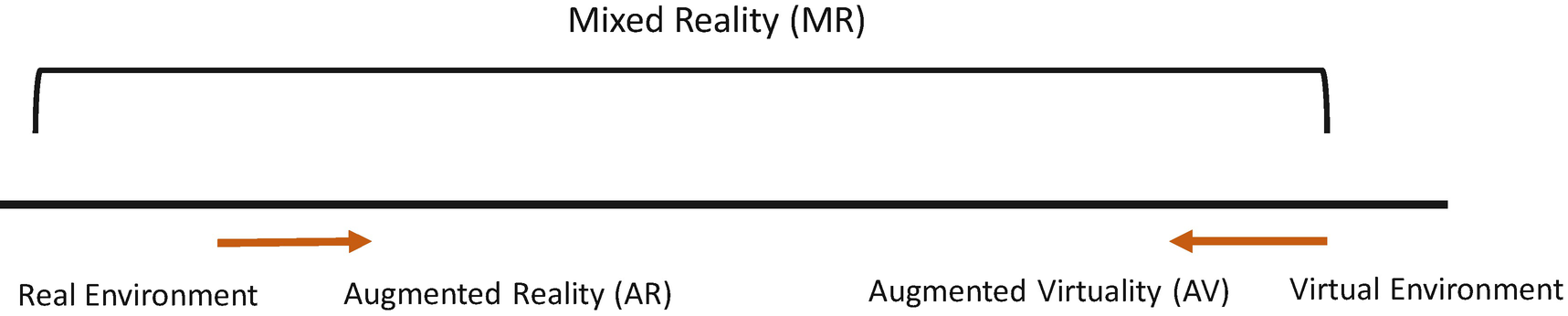
5G helps you to get an immersive experience with XR in order to transform the way you consume and interact with content, streaming experiences, etc. XR requires a high-bandwidth network. Figure [20-3](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_20_Chapter.xhtml#Fig3) provides the details of how 5G is used to provide an immersive experience.



***Figure 20-3***

XR with 5G network

* **Augmented Reality**
* In AR, virtual information and objects are overlaid on the real world. This experience enhances the real world with digital details like images, text, and animation. It goes a step beyond 2D and blends the physical and digital worlds with interactive 3D and spatially aware digital content and holograms. You can still have a grasp of the physical world around you but can see 3D digital content. You can experience all this via AR glasses or screens.
* **Virtual Reality**
* VR is a fully immersive experience where users have an in-the-moment sense of presence in a computer-generated environment. You can interact with highly convincing imagery and digital content with no direct connection to the real world. You must use a VR headset to get a 360-degree view of the world.
* **Mixed Reality**
* Digital and real-world objects co-exist and can interact with one another in real time. As shown in Figure [20-4](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_20_Chapter.xhtml#Fig4), this is a hybrid reality technique. It requires an MR headset and more power than AR and VR. It allows users to visualize 2D or 3D digital content. The technology takes into account the 3D depth map of the environment to allow hologram occlusion (hiding behind real objects) and hologram collision (interacting with real 3D objects). You can find more details at [https://www.accenture.com/us-en/services/technology/extended-reality](http://www.accenture.com/us-en/services/technology/extended-reality), <https://www.qualcomm.com/research/extended-reality>, etc.



***Figure 20-4***

Reality virtuality existence

These types of reality have been gaining traction during the COVID-19 pandemic, especially in the retail sector, defense training, health sector, etc. With them, you are able to shop general retail, make home purchases, etc. According to a survey, the XR market is expected to reach $209 billion.

**Edge Computing**

Edge computing is a networking model focused on bringing computing as close to the data as possible to reduce latency and bandwidth use. It optimizes Internet devices and Kubernetes-based and non-Kubernetes web applications by deploying computing power closer to the source of the data or instrument like drones, IoT sensors, etc. This minimizes the need for long-distance communications between client and server, which reduces latency and bandwidth usage.

Hardware and services of edge computing are a local source of processing and storage for many of these systems. An edge gateway near the source hardware processes the data and sends only the relevant data to the back-end cloud in near real time or batches, depending on the nature of the source system.

The edge device can be anything; it can be an IoT sensor, a chip in your mobile phone or personal computer, a drone, a security camera, or an Internet-connected home appliance like a refrigerator or washing machine. The edge gateway in all these sourcing systems is considered as an edge device.

Without edge computing, an online photo verification or eye scanning would be required to run the algorithm through the cloud by sending all these details over the Internet; this creates huge latency.

There are various companies already working on edge gateways like AWS, Google, Azure, NVIDIA, etc.

For example, an IoT edge provides device connectivity and analytics to physical assets in the factory IT hub environment within the digital twin environment. The data stored in the factory IT hub with the help of IoT edge requires further immediate processing and analysis. The edge IT hub sits in the same facility as physical assets with sensors, because IoT data easily eats up network bandwidth and swamps your data center environment and resources. You use a machine learning algorithm at the edge to scan for anomalies that identify impeding maintenance problems that require immediate action. With the ML, you could use visualization tools and techniques to show dashboards, etc.

With this model, edge computing transforms the way data and communication handle millions of devices across the globe. According to Statists, there are around 21.5 billion interconnected devices across the globe.

**Summary**

New technologies are ushering in the cloud native environment; it is an era of global opportunities. To support this era, organizations are growing and transforming their core business, while also pivoting to take on new opportunities.

I covered just a few trends related to the cloud native environment in this chapter. You can find many more in the industry. Research institutes and consulting organizations publish trend reports every year; just follow them to understand more. From a technology perspective, refer to the ThoughtWorks technology radar; it is updated often to provide clear details for you.