IoT Applications in a Distributed Cloud

by Volodymyr Vyshko, Vasyl Akimov, and Vladyslav Branytskyi
– CEE Technology Practice

Introduction

As the number of intelligent devices grows, it's increasingly apparent that Internet of Things (IoT) technology can solve many of the problems we face today.

One problem it can solve for developers, for example, is to move IoT backend services to the cloud, enabling the technology to operate without downtime or delays and deploy resources more efficiently.

People using IoT devices worldwide can use a single geographical place (region) for backend services. However, this creates a problem as it can introduce data storage latency and poor user experience.

Solving this requires architecting an IoT cloud and distributing it to all regions close to the user.

The AWS Cloud spans 84 availability zones within 26 geographic regions worldwide, but IoT applications still need more sites to operate efficiently.

Imagine a home security system with web cameras that should detect and save abnormal movements in the house. These cameras could be spread worldwide, connecting them to a standard cloud like AWS with a classic client-server architecture. Operating this way causes delays and growing costs for data transfer across regions, though.

In this paper, you'll find an alternative in a distributed cloud architecture, the benefits of which include:

Reduced latency, with services closer to the service-recipient companies.

Simpler data governance and regulatory compliance with data in regulated on-premises data centers.

A smaller area of communication with data processing on local edge locations versus a central cloud.

Scalability and resilience.

We'll explore distributed IoT applications with cloud benefits, architectures, tools, best practices, and solution accelerators you can use to improve your organization's IoT applications strategy.

How the IoT Cloud Differs

Cloud providers offer multiple ways to use server applications with reliable methods of maintaining horizontal scalability, disaster recovery, and high availability.

For example, you could efficiently run a major banking application in the Kubernetes cloud-based cluster with hundreds of nodes and maintain efficiency.

However, IoT applications are different. As they are distributed by nature, classic cloud architectures are no longer relevant.

The IoT cloud is a distributed cloud (Image 1) with a specialized underlying infrastructure, services, virtual private cloud (VPC), and network configuration for real-time operations and IoT data processing.

For example, your IoT devices may need to communicate with each other using the MQTT protocol with an advanced level of security and encryption.

Or, you may want to group your devices in 5G/GSM wireless zones across regions.

Cloud providers give you instruments to develop such an infrastructure by using essential services, production-ready solutions, and accelerators, which we will cover later.

The IoT cloud must provide an easy way to maintain an increasing number of connected devices across different locations with a simple firmware update process.

This infrastructure should be flexible, cost-effective, and secure.

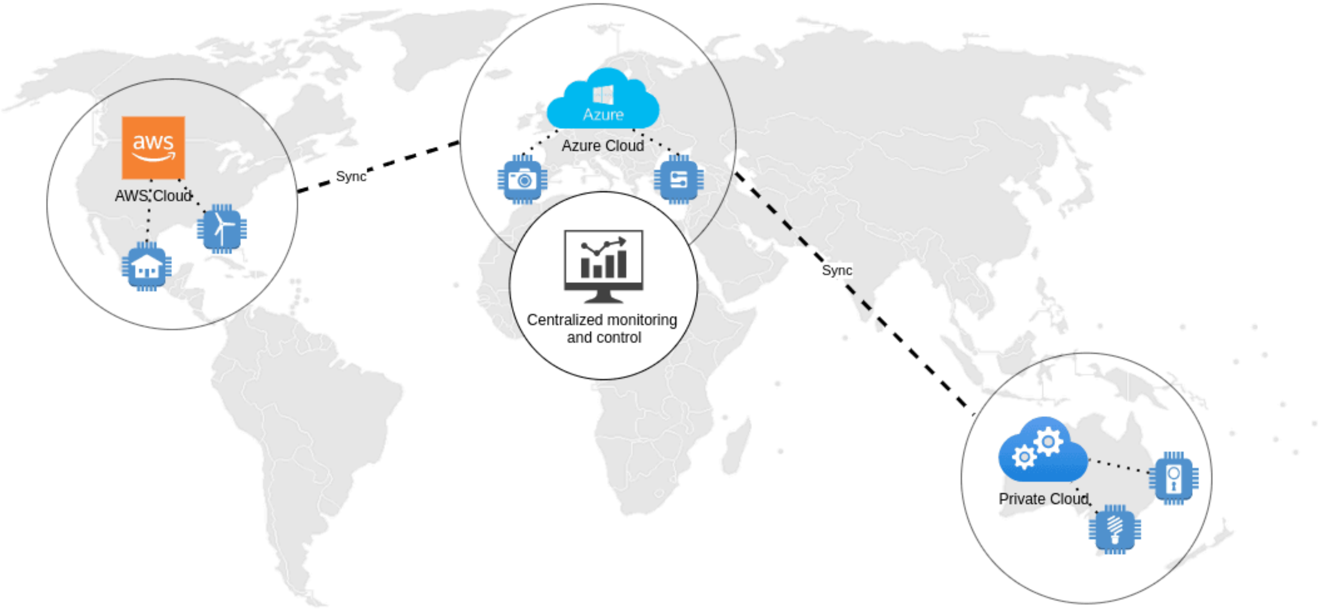


Image 1: Distributed IoT cloud

Distributed IoT Cloud Benefits

Latency

The exponential growth of IoT devices requires low latency and significant data center network throughput. Distributed cloud brings services physically closer to the service-recipient companies and reduces latency substantially.

Regulatory Compliance

Organizations often have to confirm that their applications, the infrastructure they depend on, and third-party services are certified compliant. The distributed model allows data to stay in the original location, simplifying data governance and regulatory compliance even more than a hybrid cloud.

Smaller Communication Area

A well-architected distributed IoT cloud enables users to reduce or avoid wide-area communication by using local or private networks for IoT communication and data storage. As a result, it improves overall application security and performance.

Scalability

With a distributed cloud setup, you have all of the benefits public clouds provide. Your application can be served through an edge on-premises data center or private cloud, accessing the public cloud only when necessary to meet peak demands.

Resilience

One of the problems with traditional resiliency and disaster recovery strategies on the public cloud or on-premises environment is that testing is tricky and potentially risky.

This is because tests must be carried out on live systems, sometimes at a data center-wide level.

Distributed clouds may use soft failovers with 'hot swappable' software and cloud services. Failures can be quickly and repeatedly tested in different ways.

Recommended reading: Living on the Edge – The Metamorphosis of Edge Computing

IoT Cloud Challenges

Modern high-load IoT applications with thousands of devices across multiple locations tend to introduce additional complexity to infrastructure in multiple areas.

Here are a few examples.

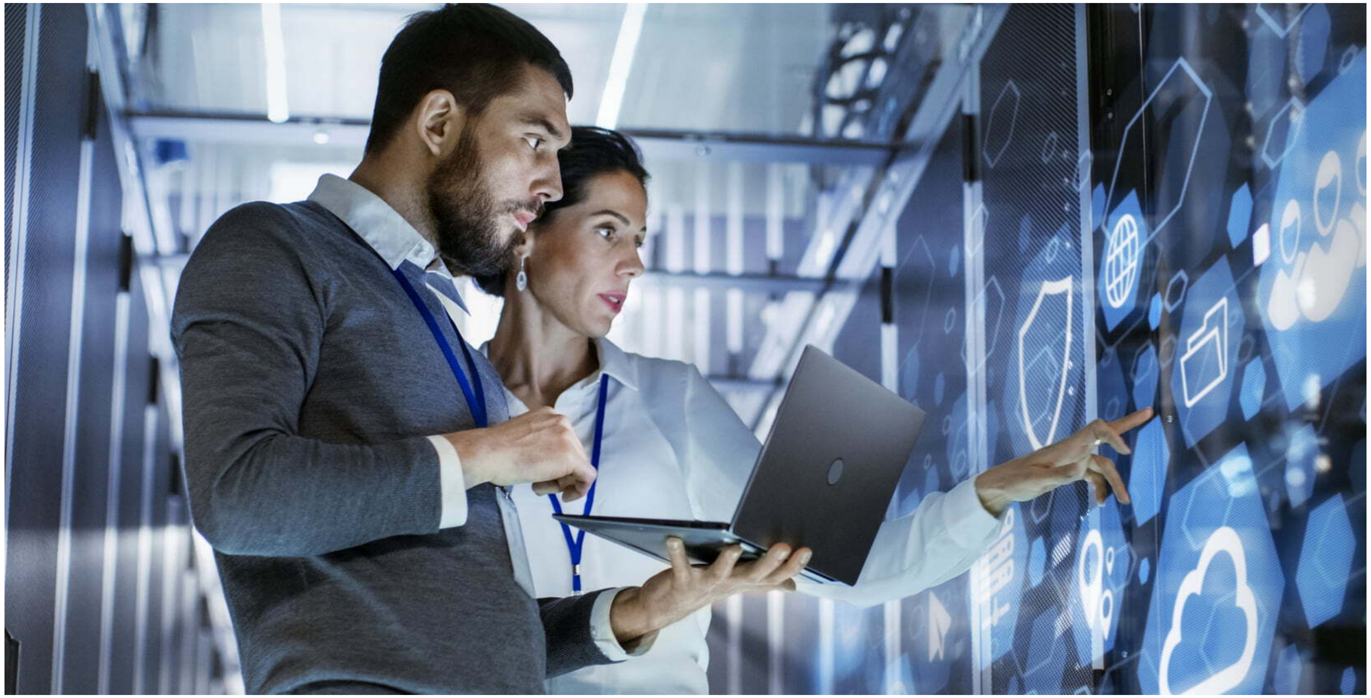
Network Configuration

An application should work across multiple regions, cloud providers, and on-premises data centers.

This requires an advanced network configuration, as a cheap VPN link between sites could be a bottleneck for the performance of the whole application.

Data Transferral

In the classic architecture, application logic is usually located on the server in the central cloud. But IoT applications tend to work with a considerable amount of data: video, photos, sensor logs, etc.



Transferring all this data to a central cloud may be a reason for high costs and low application performance.

To solve this problem, we can process the data on the local (or nearest) edge data center and transfer only relevant data to a central cloud.

Security

Security is critically important with a vast network of IoT devices.

The data passed by IoT devices should be encrypted at rest and isolated so that third-party services, IAM groups, databases, etc. cannot access it.

The physical network security of the IoT low-level subnets is crucial so there can be no unauthorized access to these devices.

IoT Device Software Updates

Classic web applications have one huge benefit: as deploying new functionality to applications is straightforward, you only need to update backend services and deploy new static files.

On other hand, IoT brings additional complexity here in that you must take care of firmware updates. Fortunately, cloud providers give you instruments to simplify the CD process for IoT devices. For example, AWS Greengrass allows for IoT firmware updates with a few clicks of a mouse.

IoT Offline Work & Data Synchronization

IoT offline work is essential.

It's not a great idea to suspend a device when losing an internet connection or stop the whole edge location with connectivity problems of the central cloud.

We must take care of the offline work and data synchronization after restoring the connection.

Unfortunately, we don’t have a silver bullet architecture here.

Instead, an architect should decide how vital data consistency is and how the reconciliation process will work for each application.

Edge Computing for IoT Cloud

Edge computing, a popular buzzword in IT, is critical in the IoT cloud context. To provide a smooth low-latency experience for the IoT user, we have to shift the computing power closer to the user. We must organize "Edge locations" where all IoT processing power can be concentrated.

Only essential data should be sent to the central cloud. For example, when building an ML-based app for detecting car accidents, you don't want to transfer all video files across the world to the central cloud where the processing cluster is placed.

Instead, you can put an ML algorithm on the edge locations, process video there, and send JSON-based events to the cloud with reports.

This is the essence of edge computing, and it suits IoT well here.

Edge computing brings additional complexity to infrastructure but offers several advantages:

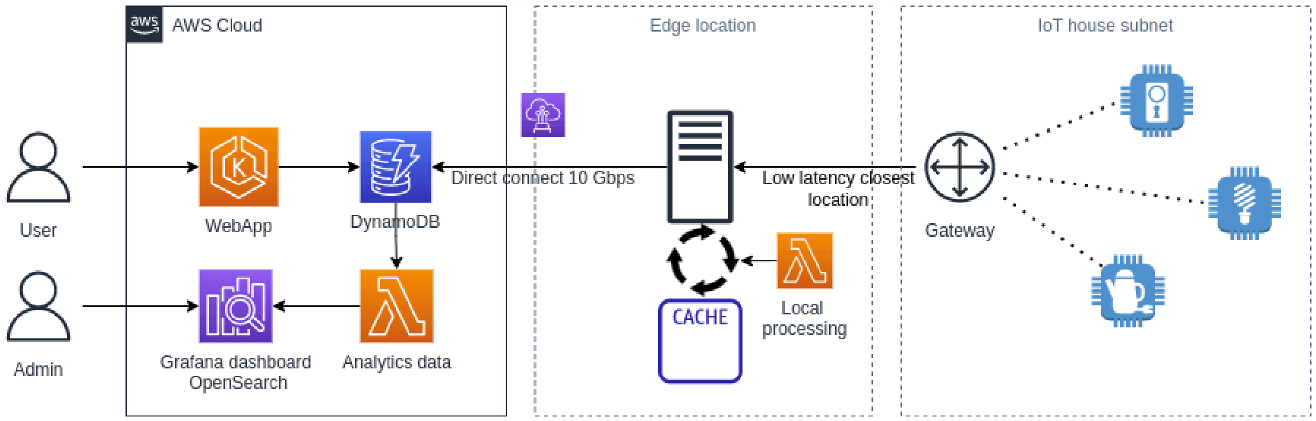


Image 2: Edge computing for IoT cloud

Cost savings. You don't have to send all the data to a cloud so you can save some bandwidth of the connection link.

Improved application responsiveness. The computing power is now located closer to IoT devices which reduces latency.

Better security. The distributed architecture of the IoT cloud with several edge locations is much harder to compromise.

Better reliability. Because of the physical separation of the edge locations, the overall system reliability becomes higher.

Of course, a distributed cloud with many edge locations is harder to maintain, especially if you combine multiple cloud providers with on-premises data centers. Such a system requires good DevOps architecture with centralized monitoring and alerting, requiring more skilled DevOps/SRE team members on the team.

Industrial IoT Platforms

The IoT is widely used in the private sector at home and has many use cases, including remote camera and security sensors access, lighting control, air quality monitoring, voice assistance, smart locks, and more.

IoT brings even more benefits to businesses, so IIoT (Industrial IoT) platforms continue to evolve to help enterprises increase their potential.

You can see a distribution of vendors providing IIoT Platforms in Image 3.

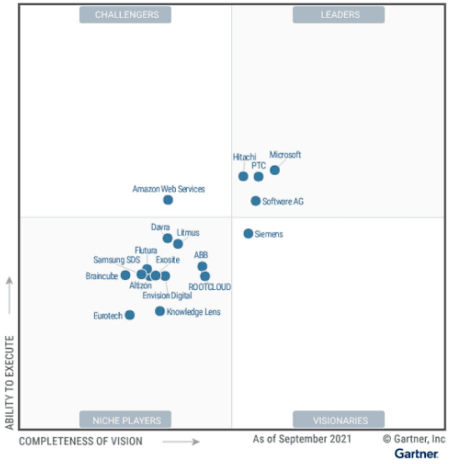


Image 3: Magic Quadrant for Industrial IoT Platforms (Gartner)

According to the Gartner Magic Quadrant, Microsoft, PTC, Hitachi, and Software AG are the most noticeable. AWS and Siemens are also worth mentioning.

But what is the IIoT Platform?

It's a set of services (and sometimes hardware) that:

- automates IIoT data aggregation,
- pushes it to a central cloud, edge location, or private data center,
- and provides tools for integration with other services like AWS S3, Redshift, or OpenSearch.

The goal of an IIoT Platform is to:

- reduce costs compared to legacy operational technologies (OT),
- simplify setup,
- and gain the benefits of cloud technologies, such as availability, scaling, DR, etc.

These platforms are extremely useful for a wide range of manufacturing, transportation, and energy enterprises, including car manufacturers and automotive businesses, food and beverage industries, metal and industrial manufacturers, chemical organizations, electric and gas facilities, transport subsectors, and many others.

More specifically, IIoT platforms provide:

Advanced device management. Services that simplify a device configuration, pairing, and cloud IoT setup.

Better cloud integration. Integration with other cloud services is simpler; for example, AWS Greengrass allows access to other AWS services through an AWS API or Lambda function via a secured communication channel.

IoT fleet analytics. A service that allows analyzing the state of your IoT fleet. It may give the health status of each device, a list of metrics, and integrate with other services for visualizing the data.

Firmware updates automation. A simplified process for maintaining continuous deployment of the firmware updates to a fleet of your devices.

High security. A secured communication channel between devices and a cloud. Fine-grained permission and access separation. Encryption keys rotation and automatic security audits.

In this paper, we cover a few such platforms that can be used for IIoT. We also provide a brief overview of related cloud services that can be used with IIoT; for example, services that allow device connection via private 5G networks by creating a distributed cloud and utilizing edge computing principles.

We must also remember that industrial companies use specific protocols to communicate with sensors; for example, DLMS/RS232 or IEC104/101 in conjunction with SCADA systems.

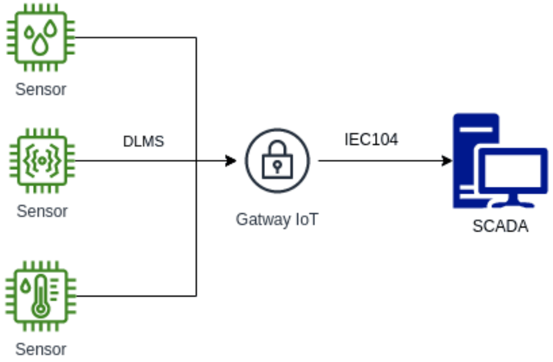


Image 4: DLMS/IEC104 private IoT

In these cases, it's still advised to connect these devices to a distributed IoT cloud. By applying a cloud connection to a proprietary system such as that shown in image 4, we benefit in several ways including cloud monitoring, OTA updates, data streaming, and backup.

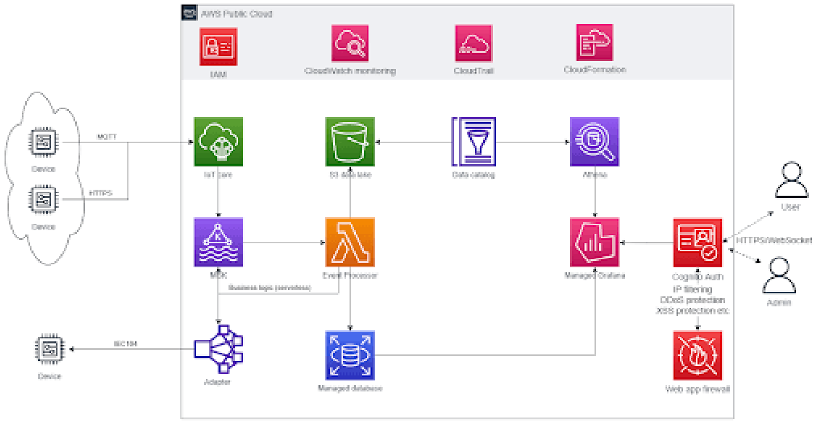


Image 5: DLMS/IEC104 AWS IoT serverless cloud solution

For example, we may use AWS IoT core to access remote IoT devices through MQTT/HTTP and a hybrid IEC104 edge adapter to make communication with a cloud through managed Kafka service.

Data from services may be processed in the cloud using serverless lambda functions and S3/Athena data sources for visualizing data in managed Grafana (image 5).

Or, we may pick another strategy and start using stateful services as shown in image 6.

In such an architecture, we put all the processing logic into the K8S cluster, which can then be easily ported into the different cloud providers or even spread across multiple providers in parallel.

We can also extract all sensitive data and store it in on-premises data centers to comply GDPR/HIPAA policies.

Still, there is one component lacking here: edge computing. It's essential to save cloud throughput and gain additional system performance and data protection. This can be achieved by AWS Greengrass or Azure IoT hub, which we'll explore in an upcoming section.

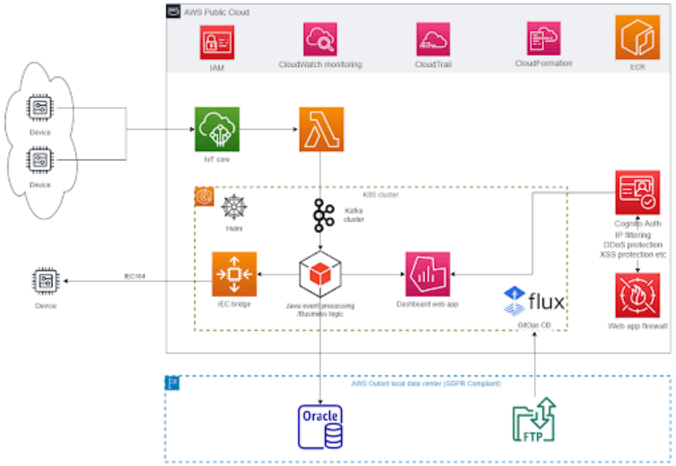


Image 6: Hybrid IoT cloud

AWS IoT and Greengrass

AWS Greengrass is an Internet of Things (IoT) open-source edge runtime and cloud service that helps you create, manage and maintain device software.

The most important part of AWS IoT Greengrass is Greengrass Core, located close to IoT devices on edge. It enables pre-processing of data from devices even when the Internet connection is slow or unstable.

Greengrass Core supports AWS Lambda functions that give the possibility to access local data, peripheral devices, ports, GPU, etc. As a result, Greengrass can gather and process data directly at the edge locations.

Such data processing enables:

- decreased data traffic to the cloud;
- working with a huge amount of devices even in places where the Internet is slow or unstable;
- saving cloud computing resources
- reacting to changes quickly avoid pre-analyzing them in the cloud.

You can see how it works in Image 7.

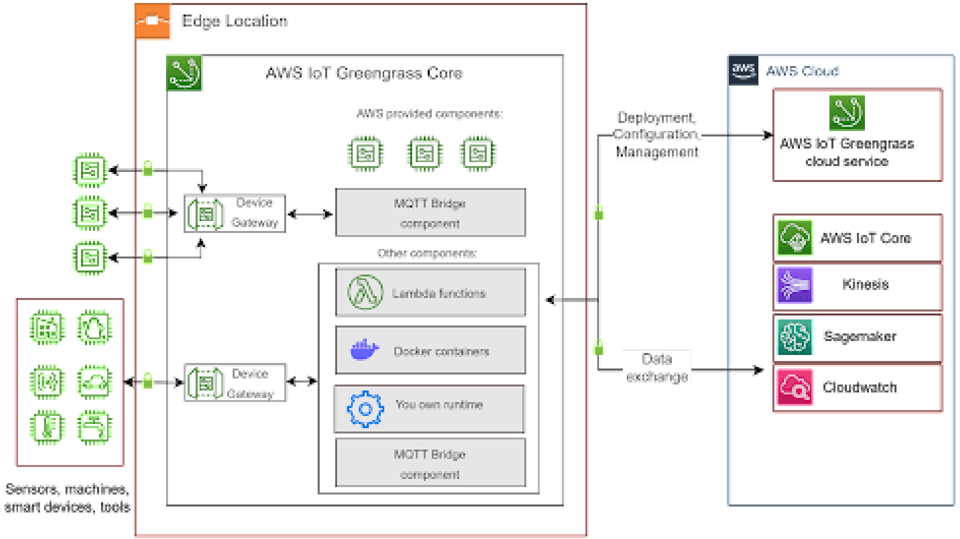


Image 7: AWS Greengrass architecture

AWS Greengrass Core provides IoT apps (components) via AWS Lambda function, Docker containers, and your runtime, which helps with local data processing, ML inference, messaging, and data management and offers pre-built components to accelerate application development.

You can create a component or use public ones prebuilt by AWS.

At the same time, Greengrass Core can act as a gateway for IoT devices and helps them to interact even if they are not connected to the Internet.

Another aspect of Greengrass is the AWS IoT Greengrass cloud service that helps to build, deploy, and manage devices. It's pre-integrated with many AWS services and can help your edge location interact with them.

Examples of AWS-provided public Greengrass components include:

- IoT SiteWise publisher, for processing data on edge.
- Log manager for uploading logs to AWS Cloudwatch
- Lambda manager, which helps to run AWS Lambda on the Greengrass device. It's responsible for managing function items and IPC.
- MQTT bridge; client devices could use it for messages both locally and with AWS IoT Core.
- Other plugins including Deep Learning Runtime, Stream data, TensorFlow runtime, etc.

AWS IoT Greengrass cloud service can help to orchestrate millions of devices. It can group them for maintenance, update, and configuration. Managing them is easy and could be done with a few buttons streaming instructions from Cloud to Edge.

Here are some examples of Greengrass usage:

Agricultural sector

The modern approach requires a massive amount of sensors, controllers of temperature, irrigation, and fertilizers.

At the same time, the remoteness of agricultural fields, the number of devices, and weak Internet connections create challenges for managing devices and quick response in case of new events, alerts, or changes.

Greengrass helps to solve these problems, even offline.

Video processing

Capturing from cameras can produce a high amount of data, and uploading everything for analysis into the cloud is expensive and time-consuming.

In some cases, it's not even possible.

Instead, this video can be handled on edge, with only the result of it being uploaded into the cloud. Greengrass ML Inference feature makes it easy to perform machine learning inference locally.

Factories

Manufacturing involves a lot of machines, tools, and humans. Production comes with many challenges including spoilage and a great deal of waste if your tool is not working as expected.

This can be prevented with sensors, monitoring, and ML algorithms that send alerts about issues before they become a problem.

AWS has many success stories of implementing Greengrass and often shares the benefits of using it. Here are just a few examples.

Hydrographic robot developer saves about 4 hours of manual labor per device.

The deployment of a new code version takes up to an hour now instead of days or weeks. AWS with Greengrass helped to develop a simulation of devices that saved money and time since now this company could test software on simulation and not real devices.

At the same time, it helped to involve other companies in the research process.

As a result, this developer saves about $1,500 per device deployment.

A Japanese company has made a great contribution to smart farming.

Accurate water volume, temperature, fan speed, and fertilizer are important for proper plant growth.

To solve this challenge, the company decided to install cameras, get images, and check the number of leaves, height, state of plants, etc. This information is handled by trained ML models.

Such an approach produces a huge amount of data and handling it on edge via Greengrass can significantly increase the speed of decision-making and save money because of the network fees.

One of the best features of Greengrass is the continuous integration and continuous deployment of the code into Edge.

The typical cases are when a new version of the application has been released, and new devices have been added to the Greengrass client devices. Both cases are the triggers of the new code delivery.

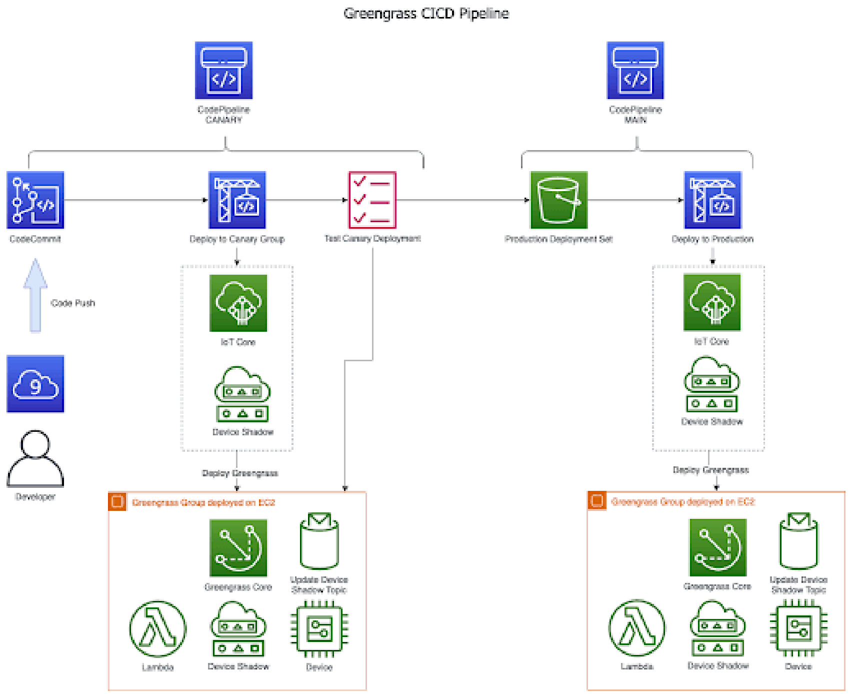


Image 8: Greengrass CI/CD diagram

Image 8 is an example (image credit: AWS) of how this could work.

In the example, you can see two paths of code deploying: CANARY and MAIN.

The canary pipeline could be used to test changes on test devices, while the main one is for deploying to production environments.

Here, we use CodeCommit as a version control system. When new code is pushed, it is taken by the canary pipeline and deployed into the test fleet.

If everything works correctly and the code is verified, it could be delivered to the production environment (main fleet).

Azure IoT Hub

Azure IoT is a collection of cloud services managed by Microsoft that allow you to manage, monitor, and control IoT devices.

Azure IoT Hub connects and manages IoT devices from edge to cloud, and Azure IoT Edge moves workloads and business logic from the cloud to edge devices.

In both cases, you can deploy and use AI and Azure services, as well as your unique business logic.

With Azure IoT Edge, you can easily tackle the challenges described previously.

IoT Offline Work & Data Synchronization

Operate your edge devices with intermittent connectivity or even offline; you can assign child devices to Edge devices so they can operate offline indefinitely. They can connect and communicate without Azure IoT Hub.

Azure IoT Edge will re-sync with IoT Hub automatically once connected to the internet.

Once connected, Azure IoT Hub retrieves the details of child devices and settings, and it updates the local cache to enable offline operation.

Security

Azure IoT platform uses certificate-based authentication so that every device, module, or actor should have a unique certificate identity. Integrating with Azure Defender for IoT provides threat protection and security posture management.

IoT Devises Firmware Updates

Import updates and deploy them easily from Azure Portal, and monitor the deployment status. The new feature Device Update for IoT Hub enables you to deploy over-the-air updates (OTA) for your IoT devices.

There are two types of updates: image-based and package-based.

To reduce bandwidth consumption, you can use a **package-based** update that targets only a specific component or app. **Image-based** updates allow you to adopt an A/B failover model as they are replicated between pre-production and production environments.

Monitoring

Integrate Azure Monitor to remotely monitor, collect and transport metrics. You can also use device-to-cloud messages and install the [azureiotedge-metrics-collector module]() on your device so metrics will be collected and sent to Azure Monitor.

Development

There are two SDKs for working with IoT Hub:

IoT Hub service SDK, which enables you to build backend apps to manage IoT Hub, send messages, schedule jobs, send desired updates to devices, etc.

IoT Hub device SDK, which enables you to build apps that run on devices and can run on a general MPU-based computing device such as a PC, tablet, smartphone, or Raspberry Pi.

Note: IoT device SDK and IoT Hub support HTTP, MQTT and AMPQ communication protocols.

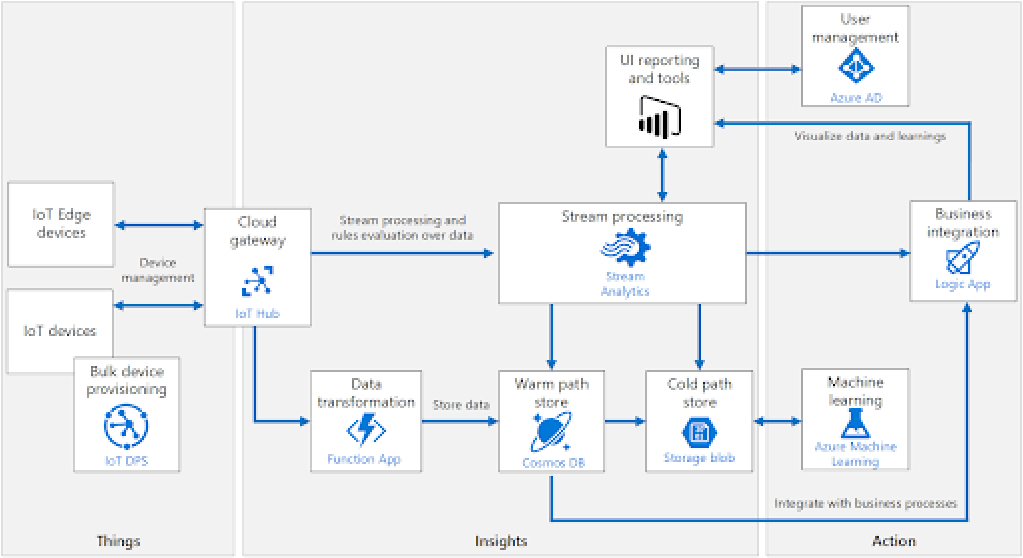


Image 9 - Example architecture with Azure IoT Hub

AWS 5G IoT Distributed Cloud Architecture

Factories, facilities, and other local networks are usually spread across one or several buildings. They could be covered with local WIFI/Bluetooth/ZigBee networks, and local networks may include wireless gateways which act as routers for wireless devices.

But sometimes it's only possible to cover some things with short-range wireless technologies.

For example, it could be reasonable to put multiple ground humidity sensors across huge fields for an agricultural company and monitor remotely when they should enable irrigation systems.

Or, the company may need to gather an air quality index across multiple forests.

In such cases, sensors and smart devices must be autonomous and communicate via a wide range of wireless technology like 5G.

One 5G base station may provide internet access for devices in a range of up to 10km.

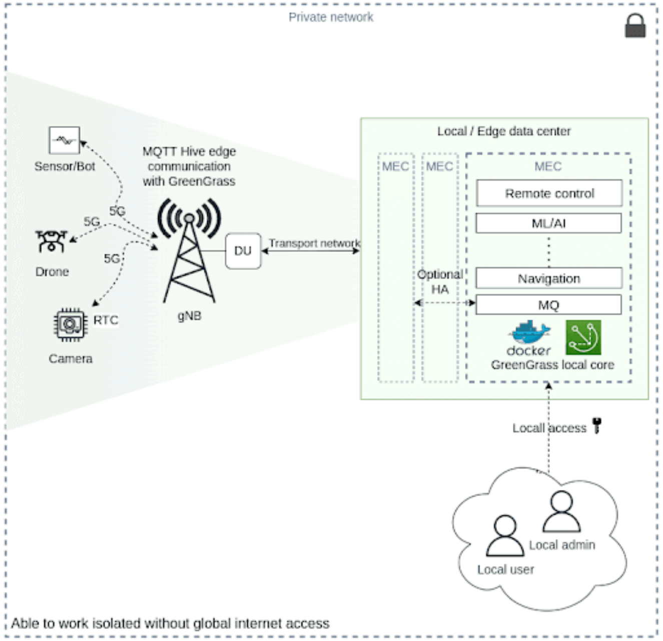


Image 10: Private 5G RAN architecture

Building a private 5G Radio Access Network (RAN) allows for a distributed IoT cloud with highly mobile devices.

This can enable you to build a wireless network that may serve an area of thousands of kilometers of devices.

This approach can be useful for multiple types of applications:

- Industrial platforms
- Smart driving
- Cloud AR gaming
- Environmental monitoring
- Security systems

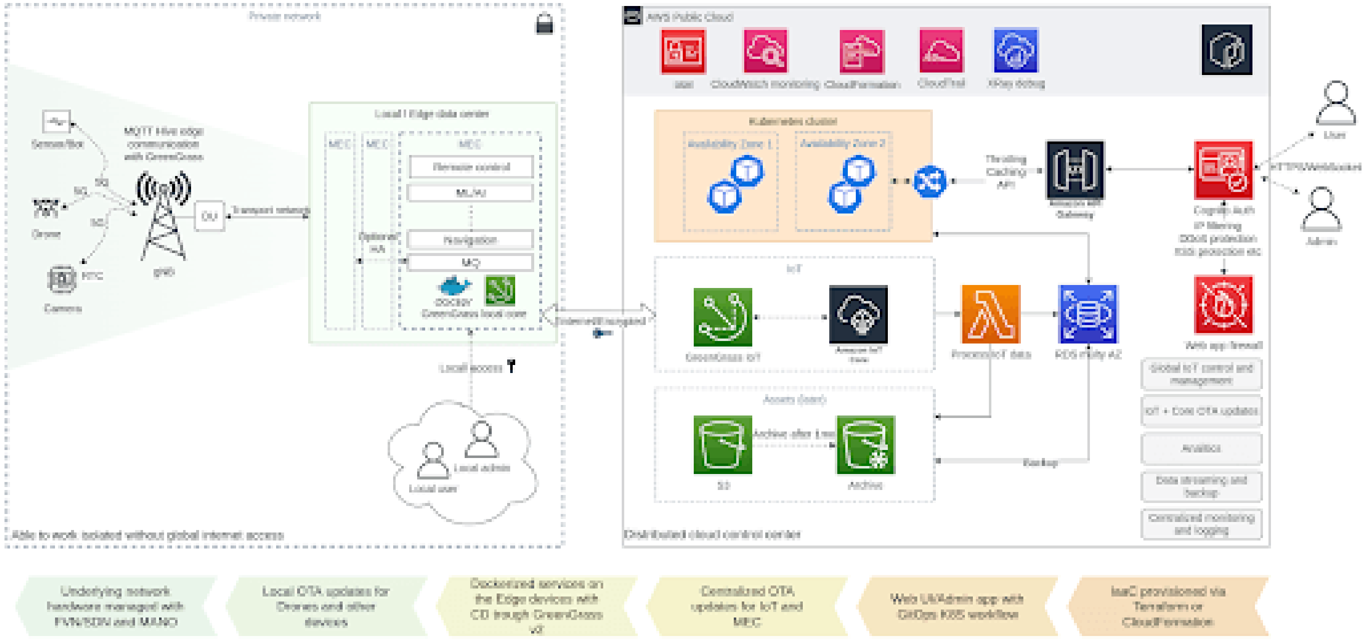
To build such a network, you will need a physical 5G RAN infrastructure connected to the cloud provider with IoT services, such as AWS Greengrass.

You can build your RAN using existing GSM/5G provider services or build the physical network infrastructure from scratch and organize a private network (Image 10).

The application is split into two parts, as in Image 11: a **private network with a MEC gateway** and a **VPC on the public cloud**.

With such an architecture, edge location can work autonomously, synchronizing the data with the cloud when a secure internet connection is up, making the application more resilient.

With Greengrass IoT, you may control all remote MEC devices, update firmware over the air (OTA) and monitor the status of the whole park of devices, do data backup and visualize it with the grafana or alternatives.



Private network
Sensor/Bot
MQTT Hive edge communication with GreenGrass
5G
5G
5G
Drone
RTC
Camera
gNB
DU
Transport network
Local / Edge data center
MEC
MEC
MEC
Remote control
ML/AI
Navigation
MQ
Optional HA
Docker GreenGrass local core
Local access
Local admin
Local user
Able to work isolated without global internet access
Internet/Encrypted
AWS Public Cloud
IAM
CloudWatch monitoring
CloudFormation
CloudTrail
XRay debug
Kubernetes cluster
Availability Zone 1
Availability Zone 2
Throttling Caching API
Amazon API Gateway
Cognito Auth IP filtering DDoS protection XSS protection etc
User
HTTPS/WebSocket
Admin
IoT
GreenGrass IoT
Amazon IoT Core
Process IoT data
RDS multy AZ
Web app firewall
Global IoT control and management
IoT + Core OTA updates
Analitics
Data streaming and backup
Centralized monitoring and logging
Assets (bare)
Archive after 1 mo
S3
Archive
Backup
Distributed cloud control center
Underlying network hardware managed with FVN/SDN and MANO
Local OTA updates for Drones and other devices
Dockerized services on the Edge devices with CD trough GreenGrass v2
Centralized OTA updates for IoT and MEC
Web UI/Admin app with GitOps K8S workflow
IaaC provisioned via Terraform or CloudFormation

Image 11 – 5G Distributed cloud physical infrastructure

In Summary...

Centralized IoT cloud brought the revolution a few years back. However, alongside the exponential growth of IoT devices connected to a single cloud provider came several problems and constraints – latency, performance, regulatory compliance, and security among them.

The Distributed IoT cloud fixes these problems using edge computing principles and a distributed cloud approach. It allows us to build secure, scalable, and flexible industrial platforms.

Giants such as Microsoft, PTC, Hitachi, Software AG, and others already offer solutions to accelerate IIoT businesses.

To start using distributed cloud approach, choose your favorite public cloud provider and a respective service for IoT.

For example, you may start with AWS Greengrass to access AWS services from the edge device and connect with IoT devices through MQTT.

Or, you might build a private 5G wireless IoT network with the AWS Greengrass (or alternatives) and a private 5G RAN infrastructure. This would enable you to serve huge industrial areas without the additional hassle of a wired network setup.

Want to evaluate your options with an experienced digital engineering partner?

How Can We Help You?

Click here to email GlobalLogic and let's explore the possibilities for your business together.

About the Authors

Volodymyr Vyshko has played different roles over his 10 years in IT, including java backend dev, js frontend, node.js backend, scrum master, lead, and cloud architect. As a trainer, he teaches courses including "Algorithms and data structures" and "Distributed systems design."

Vladyslav Branytskyi is a software engineer with experience in Computer Science, Deep Learning, and Big Data. A PhD candidate in Artificial Intelligence, he is passionate about learning new things and innovative technologies.

Vasyl Akimov, an engineer with strong system administrator experience in the Linux environment, has been working in IT for about 10 years. Vasyl's current interests and areas of specialization include DevOps methodologies, telco domain technologies, and security.

More Recommended Resources

Explore a few examples of how the cloud is making the product and platform innovations an exceptional reality.

See how AI improves project planning and helps PMs with real-time by following rule-based workflows.

Learn how to instill predictability in digital transformation programs and use the provided templates to ensure success.

About GlobalLogic

A Hitachi Group Company

About GlobalLogic

GlobalLogic is a leader in digital engineering.

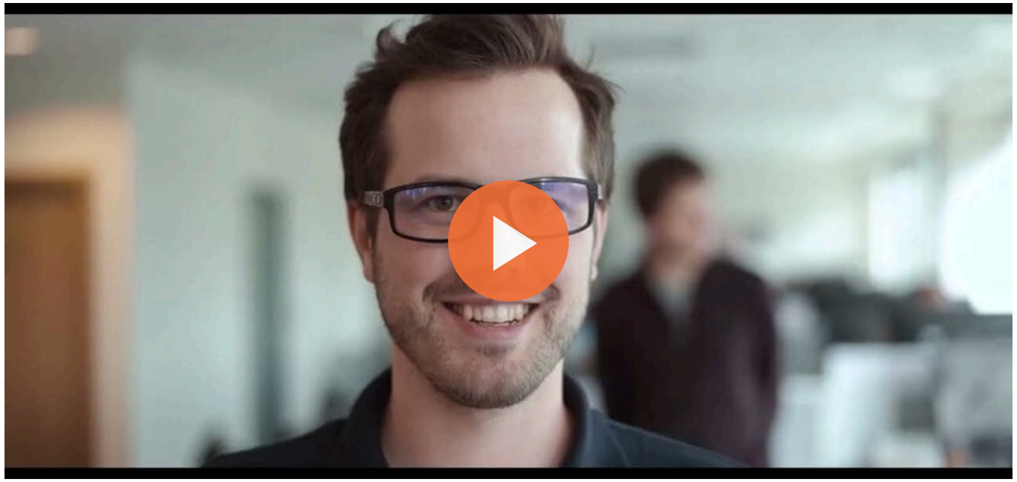
We help brands across the globe design and build innovative products, platforms, and digital experiences for the modern world.

By integrating experience design, complex engineering, and data expertise—we help our clients imagine what's possible, and accelerate their transition into tomorrow's digital businesses.

Headquartered in Silicon Valley, GlobalLogic operates design studios and engineering centers around the world, extending our deep expertise to customers in the automotive, communications, financial services, healthcare and life sciences, manufacturing, media and entertainment, semiconductor, and technology industries.

GlobalLogic is a Hitachi Group Company operating under Hitachi, Ltd. (TSE: 6501) which contributes to a sustainable society with a higher quality of life by driving innovation through data and technology as the Social Innovation Business.

Visit our website at globallogic.com.



How Can We Help You?

Let's explore how the right digital engineering partner can help transform your business. Get in touch to book your free 15-minute initial consultation now.

Let's Book It!

Thank you for reading

IoT Applications in a Distributed Cloud

GlobalLogic, a digital engineering leader and Hitachi Group Company, helps brands design and build innovative products, platforms, and digital experiences for the modern world.