Red Hat

Red Hat Hackfest: An edge manufacturing use case

Table of contents

Preface

The purpose of this document is to provide an edge computing use case to benefit organizations that mass produce products for global consumption and seek more efficient ways to integrate their IT and plant operations, so they can minimize delays and accelerate time to market. It is particularly useful for:

- IT operations leaders
- Enterprise architects
- AppDev IT decision makers

In partnership with Intel and IBM, Red Hat Hackfest provides a forum where technology companies, system integrators, and customers can design and implement repeatable solutions to business challenges.

The use case was developed as part of the Red Hat® Hackfest program, an initiative launched in 2020, to foster an innovative environment where technology companies, system integrators, and customers can discuss business challenges and work together to design and implement repeatable solutions to solve them.

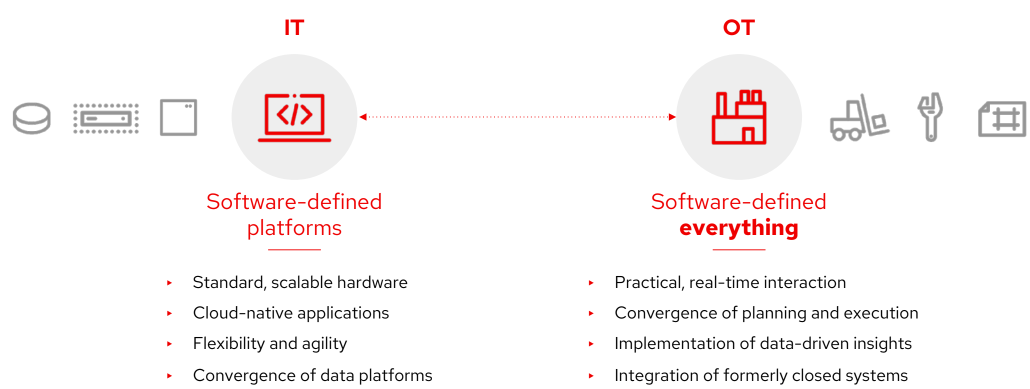
Participating companies agree on an enterprise technology stack—developed based on a specific customer challenge—and work with Red Hat and Hackfest sponsors to implement a software-defined solution on top of certified hardware and public cloud offerings. All this is done with the help of the Hackfest Community. System integrators also participate in the Red Hat Hackfest event to learn how to build and deliver a solution based on practical scenarios.

Introduction

The trend toward IT and operational technology (OT) convergence shows no sign of slowing. According to Frost & Sullivan, 68% of decision makers from industrial organizations point to IT-OT convergence as a top factor affecting their ability to meet their strategic goals.[2]

This decades-long blurring of the lines between IT and OT has helped improve how industrial organizations operate. By applying IT principles to a domain focused on machinery and plant operations, manufacturers, retailers, and energy and utility companies now have better visibility into plant-floor operations and can more easily share data-driven insights between IT and OT teams. The result is greater efficiency, improved production, and more opportunity for innovation.

Part of this ongoing convergence evolution is implementing software-defined platforms onto OT environments.



[2] Frost & Sullivan. "Edge Computing: The IT-OT convergence catalyst and enabler," July 2022.

Most industrial organizations rely on proprietary or vertically integrated OT solutions to function. This has left them with closed architectures and traditional systems that are difficult to integrate, modernize, or replace.

Following a software-defined platform approach built on open architectures, core traditional systems can be kept viable by building a seamless, horizontal, and scalable layer on top. This makes it possible for any industrial organization to benefit from the latest digital technologies and support modern use cases and business models without having to undertake significant and costly manual intervention.

The benefits and challenges of a software-defined approach

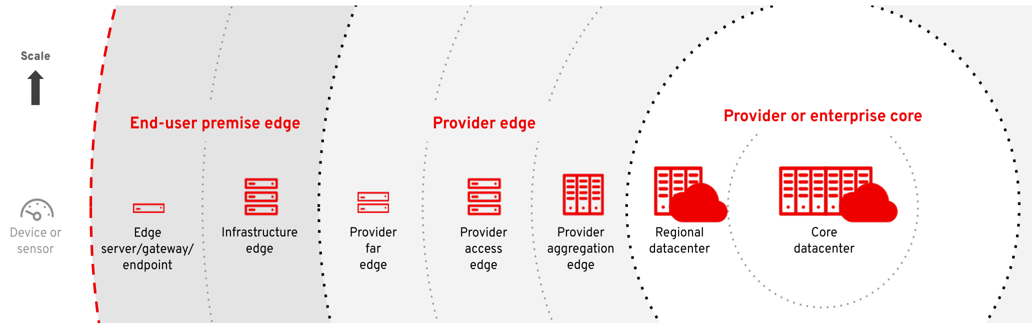
To get the most from a software-defined platform approach, The Red Hat Hackfest technical team proposes a unified, controllable, and common platform that extends from the network core to the edge and is why edge computing has become particularly relevant in industrial settings. When coupled with hybrid cloud infrastructure and 5G, edge computing can help:

- Support artificial intelligence and machine learning (AI/ML) to power data collection and analysis and improve supply chain predictability.
- Automate factory floor processes through machine-to-machine communication.
- Lower operating costs and improve productivity.

Even with all the opportunities edge computing offers, organizations must be prepared for the complexities they are likely to face when building out an infrastructure designed to support edge use cases. These include:

- **Scalability:** Remotely managing up to hundreds of thousands of nodes and clusters is challenging enough. Managing them in locations where there is little or no IT staff on hand to support them makes it even more challenging. Automated, centralized management is important for any organization overseeing a distributed IT landscape.
- **Interoperability**: No one vendor or system integrator can build and operate an entire edge stack. The best edge environments must allow for technology vendors, system integrators, and customers to join and work together on repeatable solutions that help customers solve their business and technical challenges. It is also critical that these providers guarantee interoperability across diverse hardware and software environments.
- **Consistency**: A third challenge is making sure edge sites can be managed in a streamlined, consistent way. Developers should not need to acquire special toolsets or skills to be productive in an edge environment.

The best way to plan an ideal, distributed, tier-based edge architecture is to view it from a flat perspective:



According to Omdia, in 2021, the industrial edge computing market had an estimated value of US$679 million. This is forecast to grow to US$1.4 billion by 2025 and US$3.6 billion by 2030.[1]

Use case: A seamless manufacturing production network

For more than 25 years, Red Hat and Intel have collaborated to help organizations anticipate and overcome the challenges with edge computing. Red Hat's open source software and edge computing expertise, plus its horizontal platform certified by the world's largest cloud providers, combined with Intel's hardware architecture and software toolsets, can help customers realize the full potential of hybrid cloud-ready edge computing.

As part of our ongoing collaboration and to extend the efforts of the Red Hat Hackfest Program, Red Hat and Intel have joined efforts with the Hackfest Community to develop a proof-of-value concept that can be used as a base for future edge computing projects. While the business logic around this implementation concept is focused on a manufacturing setting, it can be tailored to other edge architectures or project implementations across multiple verticals.

Use case overview

For this manufacturing use case scenario, a global t-shirt maker wants to create a linked network of production machines (or edge devices) across its various factory locations (or edge servers) that will help roll out a new t-shirt. All factories involved in production are overseen by a central plant manager working at the company's headquarters (the datacenter).

The t-shirts will be created following a production line model, consisting of four stages:

- Weaving
- Coloring
- Printing
- Packaging

At each production stage, the edge devices issue data capturing what has been produced, which is sent to the factory managers for validation and where a validator monitors for quality. If the validation process identifies any kind of negative outcome during the t-shirt production process, the affected t-shirts will be discarded–slowing down the global production schedule.

[1] Omdia. "Industrial Edge Compute and the Future of Automation Report - 2022 Analysis," March 2022.

Breaking down the factory edge

Factory edge: The hardware, server, and OS components

In this use case design, the edge server is responsible for managing the factory-level production process.

The hardware: Each factory's machinery is monitored by Fitlet2, a microsized, fanless PC device powered by dual and quad core Intel Apollo Lake processors. It provides 24x7 data collection and is designed to process and monitor applications. It can also withstand temperatures ranging from -40 to 185 degrees Fahrenheit. For this use case, assume that each device is equipped with a business logic that allows it to emulate an optimal t-shirt production model, provided by the plan manager. This business logic has already been implemented using cloud-native frameworks and packaged using container technologies.

"With our open hybrid cloud technologies, combined with Intel's strength in architecture and industry expertise, we aim to continue to advance Industry 4.0. We invite ICS and operational technology professionals and organizations to learn more."

Daniel Frölich
Global Principal Solution Architect for Industry 4.0, Red Hat

The operating system (OS): In any edge environment, including this use case, it is critical that the OS is lightweight, flexible, and vendor-agnostic and built to be as efficient as possible. It also needs to be designed as a cloud- and container-native solution, with the latest security principles in mind, and must also:

- Support 64-bit capabilities.
- Be fully immutable or at least modular.
- Allow users to run a container engine with minimal memory footprint.

In this case, the optimal OS is Red Hat Enterprise Linux®.

Container technology: Podman is a daemonless, open source, Linux-native tool designed to help develop and run Open Container Initiative (OCI) containers and pods. Users are not required to run an active container engine for its commands to work, and Podman is available in the standard Red Hat Enterprise Linux library, which makes it fully supported.

Machinery service: This service is responsible for emulation of the four-stage t-shirt production process. Ideally, it should be designed and implemented to have the smallest possible memory footprint and perform transactions with the factory floor application programming interface (API) in the shortest possible time. This will help dramatically improve performance and help ensure full compatibility with OCI standards and the Podman container engine.

The edge server: In many industrial settings, each edge server may be responsible for handling multiple workloads in remote settings where space is at a premium. Size and reliability are critical. It must also be ready to work "as usual" even in the event of network connectivity disruptions. In this specific use case, the edge server must be capable of handling multiple production models, validating conveyor belt outputs, and immediately capturing and forwarding telemetry data to the datacenter.

In this instance, the edge server is based on the Intel NUC 11th generation performance kit - NUC11TNBv7, certified by Red Hat and Intel. The server will run single node Red Hat OpenShift®.

Factory edge: Software components

With the hardware, OS, and other foundational layers in place, it is time to focus on the software infrastructure, which forms the business logic skeleton, providing integration, validation, and storage components.

Broker service: Based on the Apache Active MQ Artemis project, Red Hat AMQ supports a massively scalable, distributed, and high-performance data streaming platform. Acting as the broker service, AMQ exposes the endpoint for the production telemetry coming from the edge device machines. It even creates an individual telemetry queue for each machine awaiting validation responses.

Certificate management: cert-manager automates certificate management in cloud native environments and helps with the implementation of dynamic certificate provisioning for edge devices. It also builds on top of Kubernetes, making it possible for users to provide "certificates as a service" to developers working in a Kubernetes cluster. In this particular use case design, the issuer provides service or runtime certificates to all devices managed by a single factory in a few simple steps:

1. The factory manager completes that production plant's registration.
2. The factory manager receives a unique certificate in the form of a delegate.
3. The certificate manager then instigates a delegate certificate, using the unique certification already assigned as the "root."
4. Finally, a runtime certificate is issued to each of the factory's edge devices.

Storage: For SQL storage, PostgreSQL provides a leading open source, object-relational database system, storing non-time series data. For noSQL storage, which is especially helpful when it comes to the storing, querying, and retrieval of structured data sets, MongoDB and its document data model naturally support JavaScript Object Notation (JSON), making it simple for developers to learn and use. Functionality such as automatic failover, horizontal scaling, and the ability to assign data to location are already built in.

Factory edge: Business logic

In this t-shirt manufacturing use case, all business logic is implemented on top of the Quarkus framework, running in native mode in a container environment. This includes a number of services:

Registration service: This service is responsible for the certificate provisioning and maintenance that connects to other layers of the edge architecture. To achieve this, the registration service implements a facade between the layer manager (in this case, the factory manager) and the cert-manager services, supported by the fabric9 Kubernetes plugin.

Facility manager service: The facility manager provides the core service at the factory layer of the edge architecture. It subscribes to the central subscription system at the datacenter, providing a unique name and serial number. In return, it receives a universally unique identifier (UUID) that each and every service running on the factory layer will use when connecting to services running at the datacenter.

Product line service: This service is responsible for transforming specifications for the new line of t-shirts into a model that can be shared and understood by the production machinery. That transformation phase consists of several simple steps:

1. Collecting the product line data fed to it from the datacenter.
2. Applying margins to the basic values contained in the product line data.
3. Storing the outcome of the previous production phase in the SQL database for future distribution.

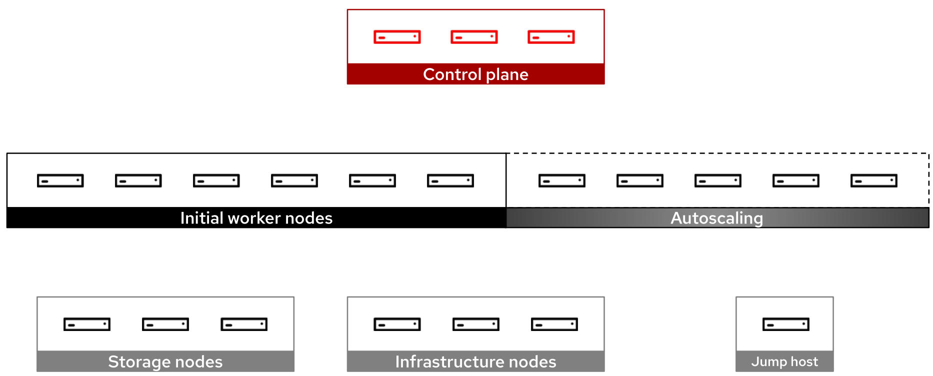
- Validating the outcome of each and every stage of machine production, relying on the product line service to gather information and communicate it asynchronously with each machine through a broker service.

Breaking down the datacenter edge

The datacenter layer of this use case is made up of several services and business logic elements, all designed to manage, store, aggregate, and retrieve data.

Datacenter edge: Hardware components

To meet scalability requirements and support the Red Hat OpenShift container platform, a cloud-based infrastructure is required, based on the following architecture:



The control plane consists of three servers with eight CPUs, featuring 25GB of RAM. The three worker nodes feature 12 CPUs and 64GB of RAM. The storage server ensures data is kept safely and consistently. The Red Hat edge platform ensures vertical scalability, and using Red Hat OpenShift as the platform ensures horizontal scalability through autoscaling features embedded in the container platform.

Datacenter edge: Software components

Once again, the software layer provides the business logic skeleton, ensuring integration, validation, and storage components.

Stream service: This use case necessitates the streaming of large volumes of messages. Basing the stream services on Strimzi and Apache Kafka helps to ensure the scalability and reliability of message flow management.

Storage: Time series database engines are a leading choice for storing Internet of Things (IoT) device data in most edge computing models. For this use case, Red Hat and Intel have picked Influxdb, which bundles dashboards, queries, tasks, and agents in one place, and helps deliver scalability, reliability, and performance needed in edge industrial settings.

Once again, for SQL storage, PostgreSQLis the recommended choice, as is MongoDB for noSQ storage.

Certificate management is facilitated again by cert-manager.

Datacenter edge: Business logic components

All business logic, including the registration and plant manager service, is also implemented on top of the Quarkus framework.

Plant manager service: Responsible for managing the overall production landscape.

- Factories subscribe to the plant manager and obtain a UUID and a trust store containing both the certificates for the connection to the datacenter and from the machines.
- Production machines subscribe to the facility manager service, which in turn, forwards the request to the plant manager service. They obtain a UUID and a trust store containing the certificates to connect to the services exposed by the factory layer.

Global product line service: Responsible for managing the product line models, users can either:

- Generate a new random global product line, or
- Receive one from the graphical user interface (or GUI, not yet implemented) through a representational state transfer REST API.

In either scenario, the product line model is validated and stored into the SQL storage.

Event collector service: Responsible for consuming the telemetry data coming from the stream service. The consumed payload gets validated, eventually decorated or adapted, and finally stored into the time series database.

Conclusion

This edge use case implemented by the Red Hat Hackfest community has shown how organizations can take advantage of Red Hat Edge, Intel hardware and expertise, and our community of open source and software partners. Following this approach, organizations can layer standardized, flexible IT capabilities over OT environments that extend from core datacenters to edge environments in a way that minimizes complexity and costs.

With a common foundation in place, organizations are in a much better position to embrace new digital solutions and innovations at scale.

Next Step

About Red Hat Hackfest

The Red Hat Hackfest program is an initiative that allows technology companies, system integrators, and customers to discuss business challenges and work together with the Hackfest Community to design and implement repeatable solutions to solve them.

How to participate

Interested in partnering with Red Hat and its technology partners to build and implement customer-centric solutions for a future Hackfest event? Please contact Red Hat Hackfest lead Andrea Battaglia or the Red Hat Hackfest team.

Glossary

This glossary includes a list of architecture components (in alphabetical order) reference throughout the document.

- **Containers:** Linux containers are technologies that allow you to package and isolate applications with their entire runtime environment—all of the files necessary to run. This makes it easy to move the containerized application between environments (such as test, production, etc.).
- **Edge computing:** This refers to computing that takes place at, or near, the physical location of either the user or the source of the data. By placing computing services closer to these locations, users benefit from faster, more reliable services while companies benefit from the flexibility of hybrid cloud computing.
- **JSON:** Short for JavaScript Object Notation, it is a lightweight format for storing and transporting data, often used when data is sent from a server to a web page.
- **Kubernetes:** Often referred to as "kube," it is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.
- **NoSQL:** These database systems are an alternative to the mainstream relational DBMS. They don't use a relational data model and typically have no SQL interface.
- **Open Container Initiative (OCI):** Established in 2015, OCI is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes.
- **Operational technology (OT):** This refers to the practice of using hardware and software to control industrial equipment, and it primarily interacts with the physical world. OT includes industrial control systems (ICSs) such as programmable logic controllers (PLCs), distributed control systems (DCSs), and supervisory control and data acquisition (SCADA) systems.
- **REST API:** Also known as RESTful API, this is an application programming interface (API or web API) that conforms to the constraints of representational state transfer (or REST for short) architectural style and allows for interaction with RESTful web services.
- **SQL:** A widely accepted standard since 1986, Structured Query Language lets users access and manipulate databases in a variety of ways.
- **Time series database management system:** This refers to a system optimized for handling time series data, as each entry is associated with a timestamp. It is designed to efficiently collect, store, and query various time series with high transaction volumes.
- **UUID:** Short for Universal Unique Identifier, it is a 128-bit value used to uniquely identify an object or entity on the internet.

About Red Hat

Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers develop cloud-native applications, integrate existing and new IT applications, and automate and manage complex environments. A trusted adviser to the Fortune 500, Red Hat provides award-winning support, training, and consulting services that bring the benefits of open innovation to any industry. Red Hat is a connective hub in a global network of enterprises, partners, and communities, helping organizations grow, transform, and prepare for the digital future.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

North America	Europe, Middle East, and Africa	Asia Pacific	Latin America
1 888 REDHAT1	00800 7334 2835	+65 6490 4200	+54 11 4329 7300
www.redhat.com	europe@redhat.com	apac@redhat.com	info-latam@redhat.com