Planning AI infusion into applications on IBM® zSystems®

About this PDF

Last updated on June 9, 2022.

Table of contents

About this PDF 2

Table of contents 3

Introduction 4

Why infuse AI into applications on IBM zSystems? 4

What options are available? 5

- **Using IBM Watson Machine Learning for z/OS** 6
- **Using IBM Operational Decision Manager with IBM Watson Machine Learning for z/OS** 9
- **Using a community-available AI framework** 10

Find more information 15

Introduction

When infusing an AI model into business applications:

1. Enterprise architects or business owners will probably have identified what new value the applications can incorporate by using an AI model. An AI model often generates additional insights and returns a score or a prediction on which transactions running as part of the application can act. If you haven't done so, identify use cases that meet your business needs.
2. The data science team within the organization should then have developed an AI model to make that prediction or return the score.

After your organization has identified the use case and the AI model, you can start planning how to infuse AI into business-critical applications. This guidance covers applications running in the following runtimes because they host most mission-critical applications on IBM® zSystems®. It is primarily aimed at application architects and application developers who will design and make application changes in the following products:

- IBM CICS® Transaction Server for z/OS® (CICS TS, also referred to as CICS in the following text)
- IBM IMS Transaction Manager for z/OS (IMS TM, also referred to as IMS in the following text), covering both IMS TM standalone systems, and IMS Database Manager and IMS Transaction Manager systems
- IBM WebSphere® Application Server for z/OS–WebSphere traditional and WebSphere Liberty (referred to as WebSphere in the following text to mean both WebSphere traditional and WebSphere Liberty)
- z/Transaction Processing Facility (z/TPF)

Why infuse AI into applications on IBM zSystems?

Infusing AI is about being able to apply AI across your enterprise, drawing on predictions, automation, and optimization to improve your business decisions and outcomes. It is also about making AI part of your day-to-day operations as part of your business processing.

By infusing AI models into applications running on IBM zSystems, you enable real-time decision making within the transactions, significantly reduce latency over making calls off-platform, and avoid the need for the data to leave the platform. The data that provides input to AI models is often relevant only at the time the transaction is being processed. For example, should this customer be approved for a loan at this time? Do the customers' current circumstances make them eligible for a better insurance rate? Is this claim fraudulent?

In addition, with the IBM Integrated Accelerator for AI in IBM z16™, you can gain more value from incorporating AI processing into runtime applications on IBM zSystems. The IBM Integrated Accelerator for AI is designed to provide extremely high performance and consistent low latency inferencing for processing transactional workloads. This opens up such possibilities as scoring every customer interaction rather than being forced to limit this to a subset and replacing (or enhancing) rules-based processing within transactions with more intelligent AI decisions. The AI accelerator offers seamless exploitation for the IBM zSystems runtimes, in that upgrades to the runtime environment should not be required, and applications that are already leveraging suitable deep-learning AI models deployed to IBM zSystems can benefit from the acceleration without change.

What options are available?

A number of options are available for incorporating AI models into transactions on IBM zSystems, and some of the considerations for which option to adopt include:

1. The type of model to be built.
2. Where the data to be input to the model comes from—directly from the application, from another source, or both.
3. What product or framework will host the model, such as IBM Watson® Machine Learning for z/OS (WMLz) or an open-source framework.
4. Where the model will be deployed–within the runtime environment, natively on z/OS, in an IBM z/OS Container Extension (zCX), or in Linux® on IBM Z®.

The first two of these considerations, the type of model and the input data, are in the domain of the data science team that builds and trains the model, which they can do using their preferred tooling either on IBM zSystems or on another platform.

This guidance explores the latter two considerations concerning where the AI model is hosted and deployed, showing some of the methods you can use to infuse AI models into your runtime environments, including CICS, IMS, WebSphere, and z/TPF. IBM zSystems is an efficient platform for hosting AI processing, and the Integrated Accelerator for AI is targeted for deep learning models, such as those generated by the IBM Deep Learning Compiler from models that use Open Neural Network Exchange (ONNX) interchange format, and can also be used for machine learning use cases.

Using IBM Watson Machine Learning for z/OS

Works well for applications running in:

CICS IMS TM WebSphere

Why use WMLz:

- You want to take advantage of WMLz and the full-function solution it offers.
- You are using models developed using Spark, Scikit-learn, PMML, XGBoost, and ARIMA.
- You are using ONNX models. WMLz can compile ONNX models for you by using the IBM Deep Learning Compiler when you import the models. If you use WMLz 2.4 and IBM z16, the ONNX models can further exploit the on-chip AI accelerator.

From the application, you can make a call that invokes an AI model deployed to the WMLz base running in z/OS or, in some cases, within the runtime itself. If you use WMLz Online Scoring Community Edition (WMLz OSCE), the model can be deployed to OSCE in zCX for scoring ONNX models. Note that the configuration using OSCE in zCX is for development or proof-of-concept purposes.

IBM WMLz supports models developed using Spark, Scikit-learn, PMML, XGBoost, and ARIMA. WMLz 2.4 will provide support for the ONNX scoring engine on native z/OS. With WMLz 2.4, deep learning models that are accessed in CICS using EXEC CICS LINK or in standalone WMLz using the REST API can exploit the IBM z16 integrated on-chip AI accelerator. When ONNX models are imported into WMLz, they are compiled using the IBM Deep Learning Compiler incorporated in WMLz.

Depending on the runtime, WMLz scoring can be invoked in the following ways:

1. Using CICS API commands: For CICS, the WMLz scoring engine can be hosted in a WebSphere Liberty server within the CICS runtime and provides a program ALNSCORE that can be invoked with EXEC CICS LINK, passing the data using CICS channels and containers. The following diagram shows an application running in CICS and calling the WMLz scoring service configured in a CICS Liberty server, using the EXEC CICS LINK command.

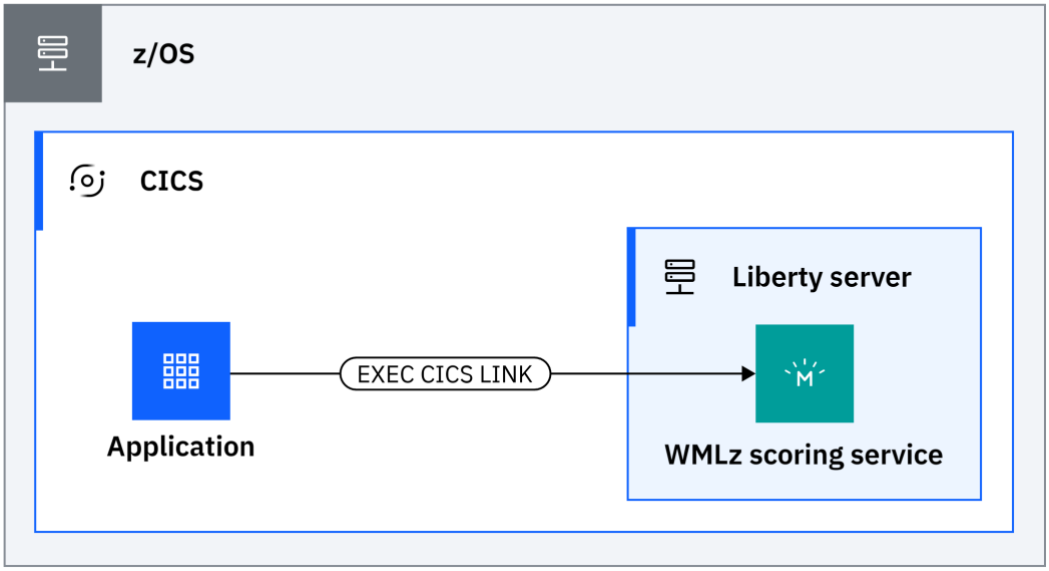


Figure 1. Invoking EXEC CICS LINK call from CICS into WMLz scoring service

2. Using Java API: For WebSphere, a Java API can be used to call the WMLz scoring feature configured in a WebSphere server. The following diagram shows an application running in WebSphere Application Server and calling the WMLz scoring service through a Java API.

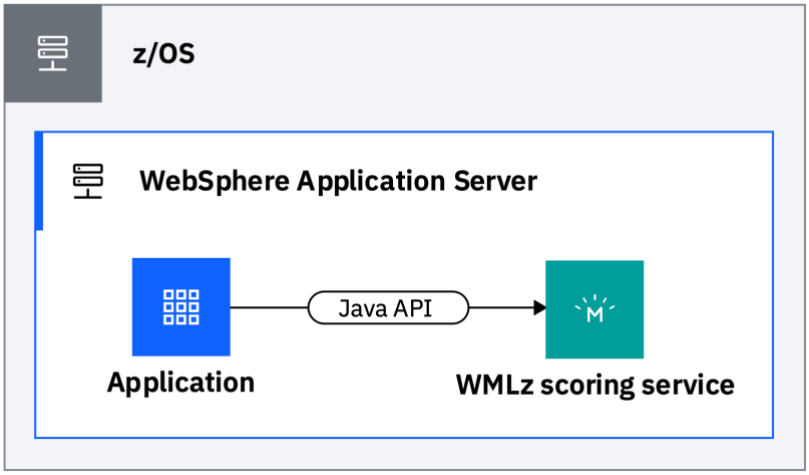
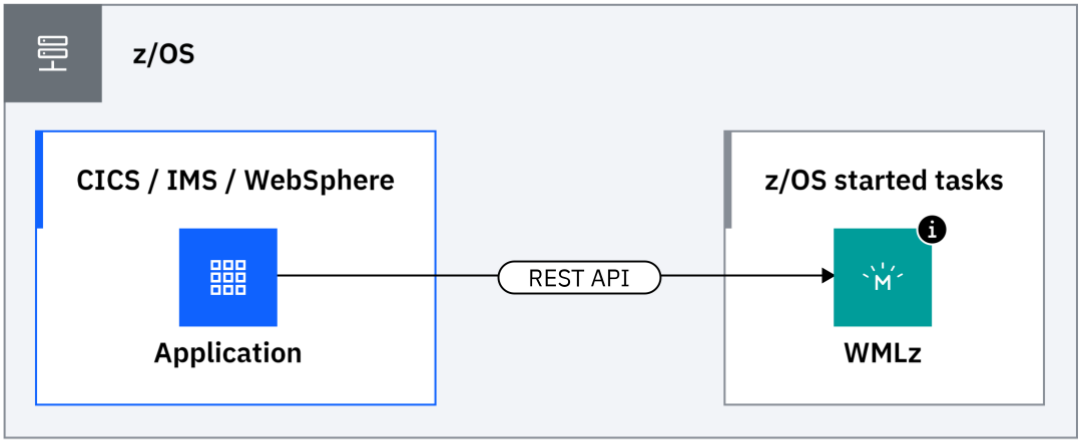


Figure 2. Invoking Java API call from WebSphere to WMLz scoring service

3. Using REST API: For IMS, and as an alternative for CICS and for WebSphere Application Server, the model can be invoked using a REST API. You can make the REST API call from the application by using IBM z/OS Connect with the API requester, or using the z/OS client web enablement toolkit. The following diagram shows an application running in the IBM zSystems runtime (CICS, IMS, or WebSphere) and calling a REST API to WMLz on z/OS.



Alternatively, you can use WMLz Online Scoring Community Edition (WMLz OSCE) and deploy it into zCX for development or proof-of-concept purposes.

Figure 3. Invoking REST API call from runtimes on z/OS to WMLz

Learn more

- Embedding the Watson Machine Learning for z/OS scoring service in a CICS region using the WMLz ALNSCORE program in WMLz documentation
- This Redpaper on Optimized Inferencing and Integration with AI on IBM Z shows a CICS application using a model in WMLz to predict a credit risk score
- The options for configuring the WMLz scoring service in a WebSphere server, either local to the application or remote from it, in WLMz documentation
- IMS applications can take advantage of the z/OS client web enablement toolkit provided with z/OS to interact with the REST APIs provided by AI models, see z/OS documentation.

Using IBM Operational Decision Manager with IBM Watson Machine Learning for z/OS

Works well for applications running in:

CICS IMS TM WebSphere

Why use ODM with WMLz:

- Your application already uses Operational Decision Manager (ODM) Rules, so the rule can be extended without application change.
- You want greater transparency into the decision logic that uses the AI result because this will be visible in the rule.
- You are using models developed using Spark, Scikit-learn, PMML, XGBoost, and ARIMA.

An ODM rule driven by the runtime application can be enhanced to reference a model deployed to WMLz, and then use the prediction from the model in the rule. ODM uses a highly efficient interface between ODM and WMLz.

Many CICS and IMS applications already use ODM rules to inform their decisions, in which case a rule called by the application can be enhanced with machine learning to drive an AI model deployed to WMLz. If the application does not currently use ODM rules, it can be updated to use an ODM rule that drives the AI model via WMLz, and hence include additional insight in the result from the rule. This use case has the added benefit of transparency in the use of the AI prediction because it's visible in the rule. In this configuration, IBM WMLz supports models developed using Spark, Scikit-learn, PMML, XGBoost, and ARIMA.

The following diagram shows an application running in the IBM zSystems runtime (CICS, IMS, or WebSphere), calling an ODM rule that uses the WMLz service via a Java API.

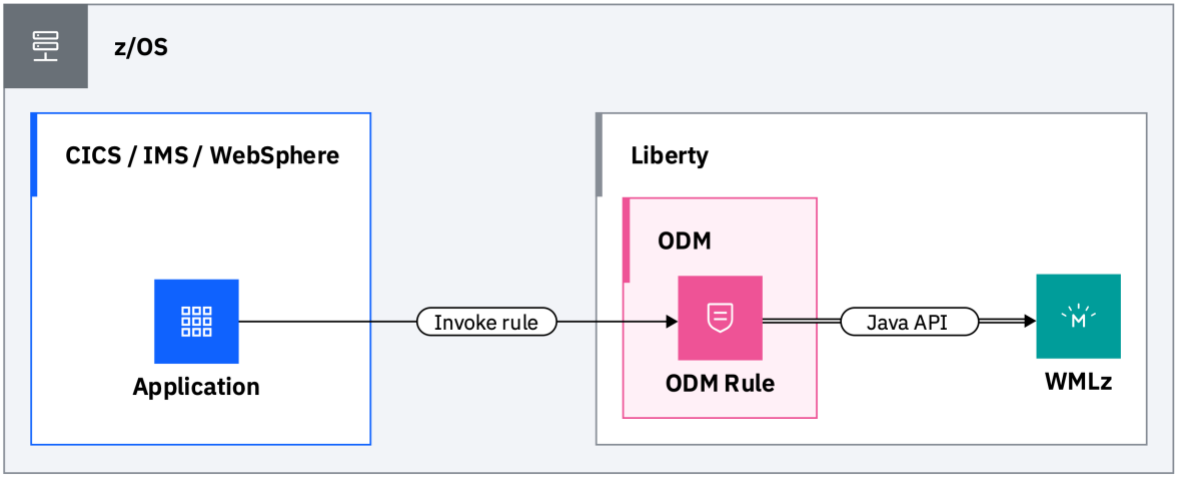


Figure 4. Using ODM with WMLz

Learn more

- Integration of WMLz with ODM in ODM documentation
- This tutorial in ODM documentation steps through enhancing ODM rules with IBM Watson Machine Learning for z/OS predictions. The ODM documentation also covers how to run the Miniloan sample application used in this tutorial as a CICS or IMS application.
- Make smarter decisions: Apply intelligence to your Z applications with digital decisioning Webinar

Using a community-available AI framework

Works well for applications running in:

CICS | IMS TM | WebSphere | z/TPF

Why use a community-available framework:

- You want to use or already have a model that uses a framework with which your data science team is familiar.
- You prefer community-available or open-source options.
- You are using IBM Snap Machine Learning (Snap ML) or TensorFlow because on IBM z16 they can exploit the on-chip AI acceleration.
- You are already using z/OS Connect or want to use it to simplify the coding in the application.

From the application, you can make a REST call to an AI model deployed in a framework such as Snap ML, TensorFlow, or PyTorch. These frameworks can be hosted in Linux on IBM Z, and alternatively, TensorFlow and PyTorch can also be hosted in a zCX instance within the z/OS environment. When using zCX, the call uses an optimized form of access within z/OS. When using Linux on IBM Z, the call can use Shared Memory Communications (SMC) for efficient access.

Models using TensorFlow, IBM Snap ML, or any model exported using the ONNX format and compiled using the IBM Deep Learning Compiler can exploit the IBM z16 integrated on-chip AI accelerator.

The REST APIs provided by the AI model can be driven from the runtimes in many ways, including:

1. Using z/OS Connect: CICS and IMS applications can take advantage of z/OS Connect and the API requester to drive the REST APIs. For IMS, you can tune this configuration by using pseudo wait-for-input (PWFI) and WFI-capable applications. The following diagrams show an application running in the IBM zSystems runtime (CICS or IMS) that uses z/OS Connect to invoke a REST API call into the AI framework, deployed either in zCX or Linux on IBM Z.

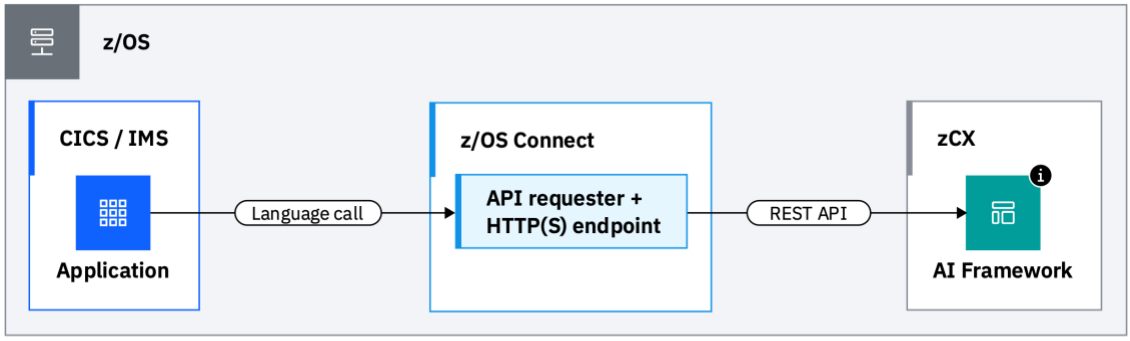


Figure 5. Invoking REST API call from runtimes on z/OS to AI framework in zCX using z/OS Connect

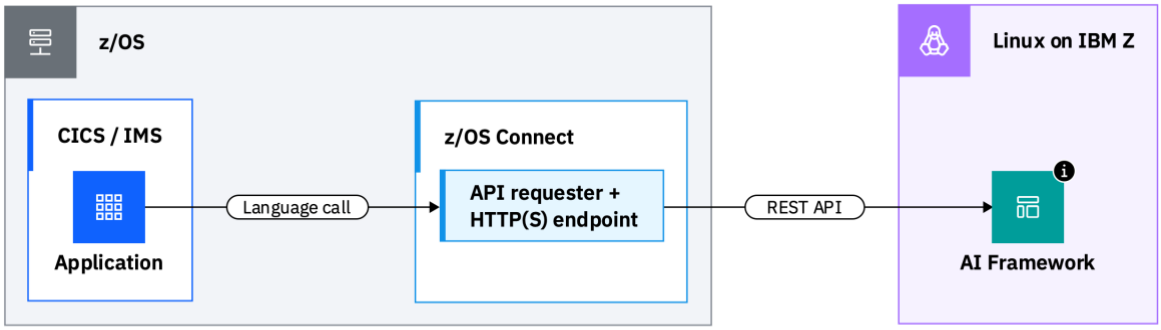


Figure 6. Invoking REST API call from runtimes on z/OS to AI framework in Linux on IBM Z using z/OS Connect

2. Using Java libraries: Java applications, including those running inside WebSphere, can easily make use of the REST interface. This can be made even easier in WebSphere Liberty or for CICS Java applications running in a CICS Liberty JVM server by using the Rest Client for MicroProfile. The following diagrams show an application running in WebSphere that invokes a REST API call into the AI framework, deployed either in IBM zCX or Linux on IBM Z.

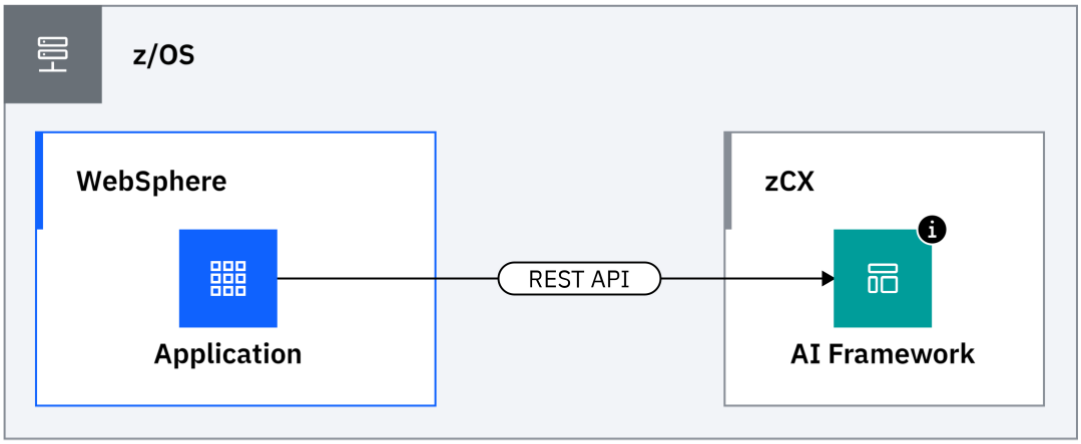


Figure 7. Invoking REST API call from WebSphere to AI framework in zCX

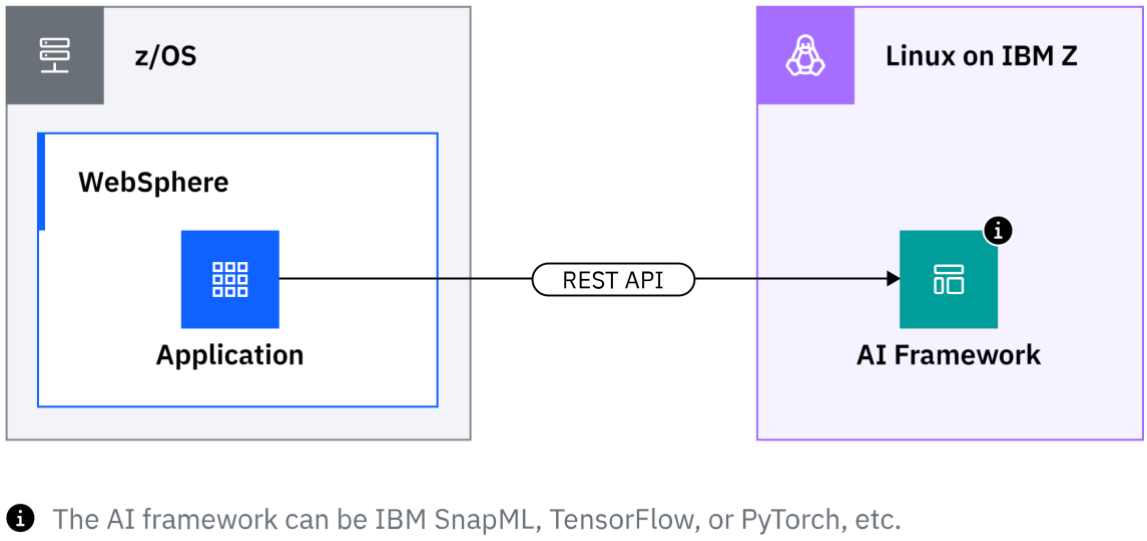


Figure 8. Invoking REST API call from WebSphere to AI framework in Linux on IBM Z

3. Using z/TPF REST consumer support: z/TPF applications can make REST calls by using z/TPF REST consumer support. The following diagram shows a z/TPF application making a REST API call to the AI framework in Linux on IBM Z.

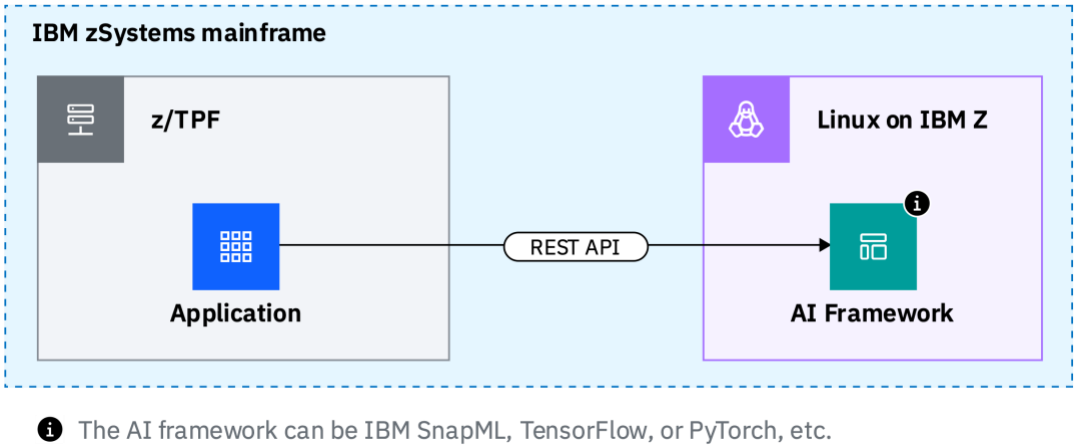


Figure 9. Invoking REST API from z/TPF to AI framework in Linux on IBM Z

4. Using CICS API commands: CICS applications can use EXEC CICS WEB commands to issue HTTP client requests.

5. Using the z/OS client web enablement toolkit: IMS applications can leverage the z/OS client web enablement toolkit, using the HTTP/HTTPS protocol enabler APIs to invoke the AI model and the JSON parser to interact with, create, or parse the corresponding JSON payloads.

Learn more

- Using z/OS Connect to drive APIs provided by AI models:
  - Calling REST APIs from z/OS applications using z/OS Connect in z/OS Connect documentation
  - Sample code showing a REST API call from IMS using z/OS Connect on GitHub
  - Tutorial showing a REST API call from CICS using z/OS Connect on GitHub
- Other options for driving REST APIs:
  - REST consumer support that enables z/TPF application to call a REST service on a remote system through the z/TPF HTTP client in z/TPF documentation
  - Making client HTTP requests using EXEC CICS WEB commands in CICS documentation
  - z/OS client web enablement toolkit documentation
  - z/OS client web enablement toolkit samples on GitHub
  - Invoking RESTful services from WebSphere Liberty using the MicroProfile Rest Client feature in WebSphere documentation

Find more information

- Journey to AI on IBM Z and LinuxONE
- Learn more about IBM z16 and the on-chip AI Acceleration
- Learn more about IBM Watson Machine Learning for z/OS v2.4

If you have any questions, feel free to let us know by contacting us.