**Overview of SAP AI Core and SAP AI Launchpad**

SAP AI Core is our own AI workload runtime, which is designed to support most of the commonly used machine learning frameworks.

Finally, management and operations of AI functions will be unified across various run times through this SAP AI Launchpad.

**SAP AI Launchpad:** is the central vehicle for SAP teams as well as customers and partners to manage their AI functions across all landscapes and deployment options.

As of now, it contains an operations manager to manage the subtenants and AI service tenants to train, deploy, operate, and monitor models in the productive environment. Here, operations managers can be considered as the main tenant and service tenant could be considered as the subtenant.

For example, if you as a partner want to bring your own customer for some of your AI services then, your customer’s AI workloads will be running in the service tenant namespace. This way we could enable tenant isolated way of executing tenant specific workloads. Other features such as content manager and functional explorer manager are in the pipeline. They are basically helping to share AI functions as content and providing options to explore and deploy in your own space.

These features are basically simplifying the whole ML complexity and enables readily available solutions for some of the commonly used AI functions.

**SAP AI Core**: is accelerating the development and productization of compliant AI functions with out-of-the-box integration into SAP Solutions. With AI Core, AI functions can be developed using any open source ML frameworks that the user prefers and fulfills their business needs. AI Core is completely compliant with AI API spec and provides full lifecycle management support, such as model training, data set management, model artifact management, deployment, and interface management.

As our AI code is built around the state-of-the-art open source solution, such as ARGO Workflow and KFServing frameworks. Therefore, in place for workflow and serving have to be ARGO and KFServing compliant. As they are open source and available in the public domain, their detailed features and specs can be offered in their portal and in their respective GitHub repositories.

Steps to create ai sample demo in Visual Studio

We created an AI Core connection then we did an admin operations we configured the git we registered the applications we registered the docker credits and then for ML operations we created a Resource Group and Resouce Group named aidemo1 and in before we registered for the ML before we expanded the ML scenario we registered the ObjectStore creds and then in the aidemo1 scenario we registered the artifact and once the artifact has been registered then we created a training configuration we did run the training after that and then once the training has been successful a model was generated which reflected in the artifact section then we used the same artifact out here the model for you in the serving and we first created a serving configuration and using the same model we created a deployment and once the deployment URL got generated we used the same deployment URL in a third party tool like Postman and we did the inferencing so this gives a total from onboarding to training and a serving experience through the AI Core Toolkit in Visual Studio.