Topic	Issue/Challenge	Context, Actions & Solution	Decision	Impact on Design
LLM Monitoring / Validation component	Possible candidate Arize Phoenix provides no authentication possibilities	• Checked for authentication possibilities in Arize Phoenix but found only the option for authentication via an Arize account • Compared with potential alternative Langfuse (which provides SSO authentication) and checked for features: from monitoring perspective everything covered	Use Langfuse instead for LLM monitoring and Ragas for LLM/RAG validation	Decision for Langfuse and Ragas over Arize Phoenix
Monitoring component	Unclear whether monitoring of classical ML models would profit from additional tools like Evidently AI or Deepchecks, or if functionality provided by Databricks is sufficient	• looked in more detail into functionality and deployment efforts for Evidently AI and Deepchecks: both seem like valid open source solutions but deploying them needs in both cases an extra monitoring component which seem not suited for managing multiple models at the same time • investigated Databricks regarding monitoring functionality: Model Serving endpoint provides resource monitoring and by using Inference Tables every model call with request and response can get stored • the model validation component of Deepchecks could get integrated by just generating reports and storing them in MLflow (like Giskard reports)	Use Databricks for monitoring over additional tools like Evidently AI or Deepchecks	Do not include and deploy Evidently AI or Deepchecks
Validation component: Giskard	Issues with running Giskard because of dependency incompatibilities (griffe)	• Manually installed older version of dependency griffe (temporary) • was soon reported in https://github.com/Giskard-AI/giskard/issues/2000 and then fixed	*None*	*None*
Platform: Databricks	Changing some deployment settings of the Azure Databricks Workspace lead to a complete redeployment	• investigated these settings and made sure to configure them early and without need to change them later on again • looked into configuring all resources inside	*None*	Higher priority on IaC and automation in regard to resources in Databricks

		Databricks via IaC (TF) as well		
Platform: Databricks	Adding a Unity Catalog to a Workspace depends on the Workspace URL which changes with every redeployment	• investigated possibility to fix or reuse Workspace URL but found no solution • Waited for attachment of Unity Catalog until Workspace configuration was stable and no redeployment was necessary anymore • Attachment of Unity Catalog was fast	Still use Unity Catalog	Poses constraint regarding automated (re)deployment of Databricks as every time the Workspace URL changes the Unity Catalog needs to get requested via ticket
Platform: Databricks	Existing compute resources in Databricks get no access to the Unity Catalog in Databricks after it is added	• investigated possibility to refresh compute resource to allow access to added Unity Catalog but found no possibility • Recreated compute resource to have access to Unity Catalog • Refrained from provisioning compute resources for users via TF and instead configured policies enabling users to easily create a compute clusters themself if needed	Do not deploy compute resources for every user via IaC but only policies for easy provisioning through user in UI	*None (as not depicted on this detail level in initial design)*
Platform: Databricks & Networking	Access of MLflow model registry in Databricks not working from inside of Notebook in Databricks after configuring IP whitelisting to Bosch IP range	• For increased security access to Databricks Workspace is limited to Bosch IP range • connection from Notebook in Databricks to MLflow model registry is getting blocked as the outgoing IP address is not whitelisted • investigated possible configurations as connection should be possible inside of network (without external traffic and getting blocked through IP whitelisting) but found no working solution • tested solution with NAT gateway and fixed IP address for outgoing traffic from Databricks networks and adding this IP address to IP whitelist and confirmed it to work	Use of NAT gateway with fixed IP address attached to Databricks networks and additional whitelisting of this IP on Databricks Workspace	Additional need of NAT Gatway with fixed outgoing IP address as part of infrastructure setup
Platform: Databricks &	Proxying Azure OpenAI instance through Databricks serving	• added Azure OpenAI model as external model to Databricks serving in order to	Do not use via Databricks serving but only with	No unified location for accessing also Databricks

Networking	component is not working because of network issues	have unified access point for all models • access to OpenAI model from Databricks fails with network/whitelisting error although IP address of NAT gateway for outgoing traffic form Databricks networks is whitelisted on Azure OpenAI deployment component • investigate issue but found no working solution without disabling whitelisting on Azure OpenAI deployment • behavior suggests that either some Databricks traffic is still going out not using the NAT gateway or that communication with other Azure services is somehow happening internally as the Databricks workspace is also (partly) an Azure managed service	direct endpoint in Azure as no working solution was found while still keeping IP whitelisting in place	external models, instead direct access to Azure OpenAI model
Automation & Secrets	No long lasting GitHub token of a technical user to deposit in Databricks to allow automated updating of Git repositories	• Databricks needs a git-credential object with a GitHub token to allow access to the repository (users can create own tokens but not feasible for automated setup) • GitHub Action of existing CI/CD processes contains GitHub token with necessary permissions but is getting invalidated after the pipeline finishes • As the git-credential object can only exist once it needs to be created in the general MLOps infra TF stack but it is needed for updating every project and cannot be changed from the TF stack of the projects • looked into creating a long living GitHub token using the GitHub TF provider and an extra GitHub App • looked into patching the git-credentials via the Databricks CLI before using it in the automation step in the pipeline	For now add additional step to Pipeline which patches the git-credentials object in Databricks with a currently valid short living token	*None*
Networking &	Understanding different deployment	• Azure Databricks Workspace has many	*None*	Increased complexity and

Security	options of Databricks Workspace and the impact on network and security as well as the general setup of the network (security rules, private endpoints) was challenging	different deployment options (for example vnet-injected) • Secure setup of Databricks and also other MLOps tools as well as access to those from applications while still having whitelisting on Bosch IP ranges in place causes need for private endpoints and advanced networking configuration		time effort for setup of network and security