



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



PaaS Workflow description

Giovanni Savarese, INFN-BA

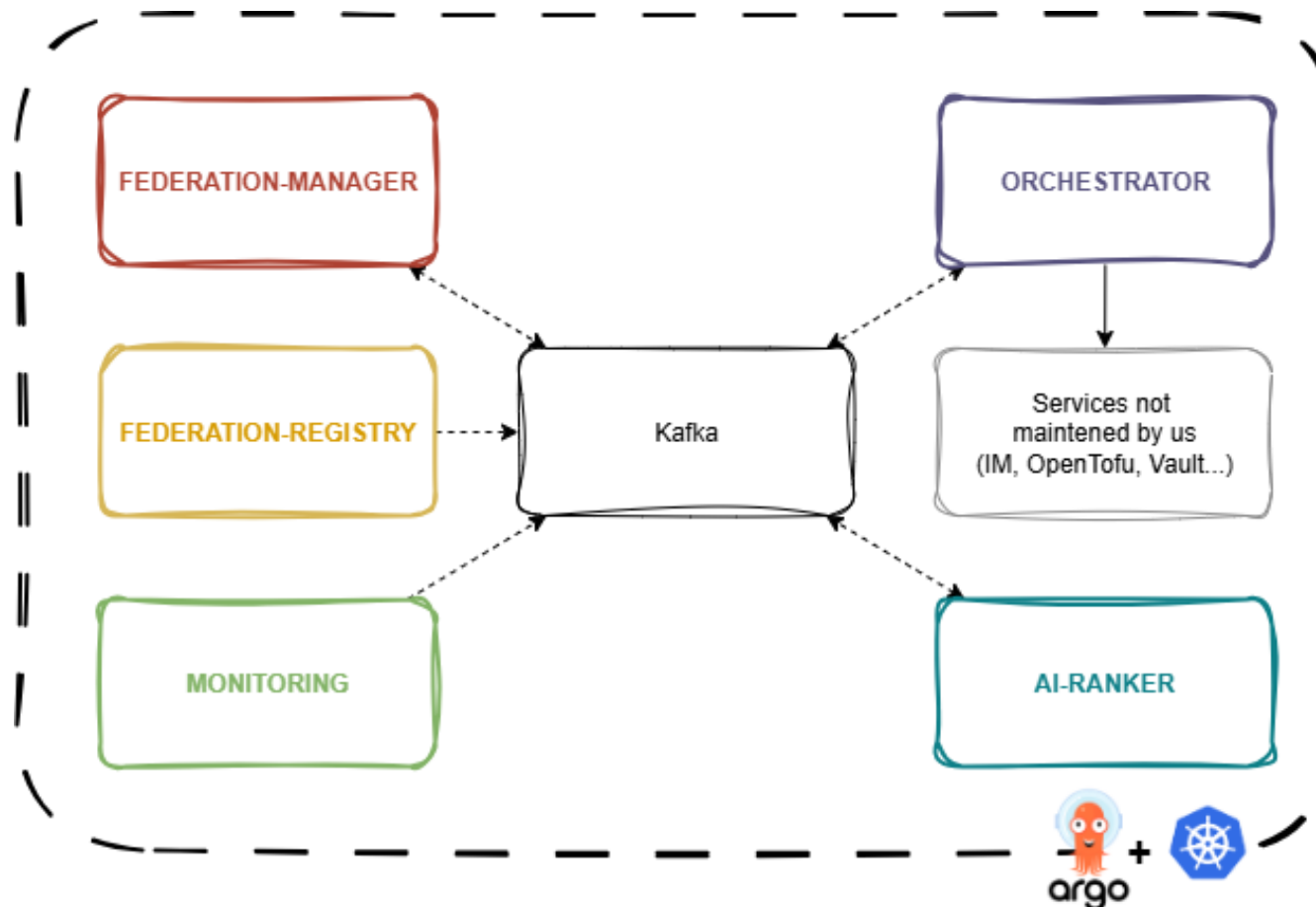
Luca Giommi, INFN-CNAF

Bari, 09.09.2025

Sviluppi PaaS, Bari, 08-10.09.2025

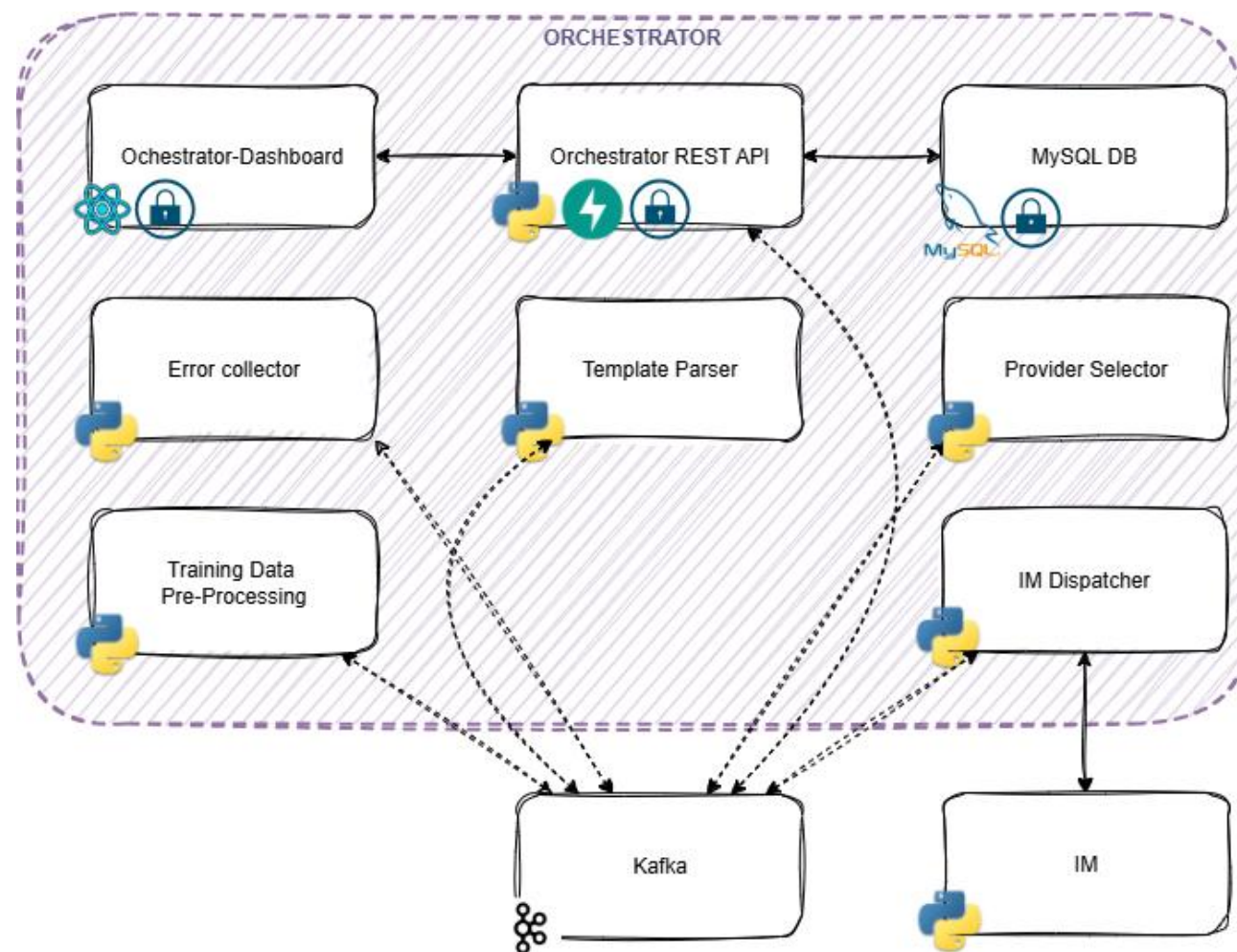
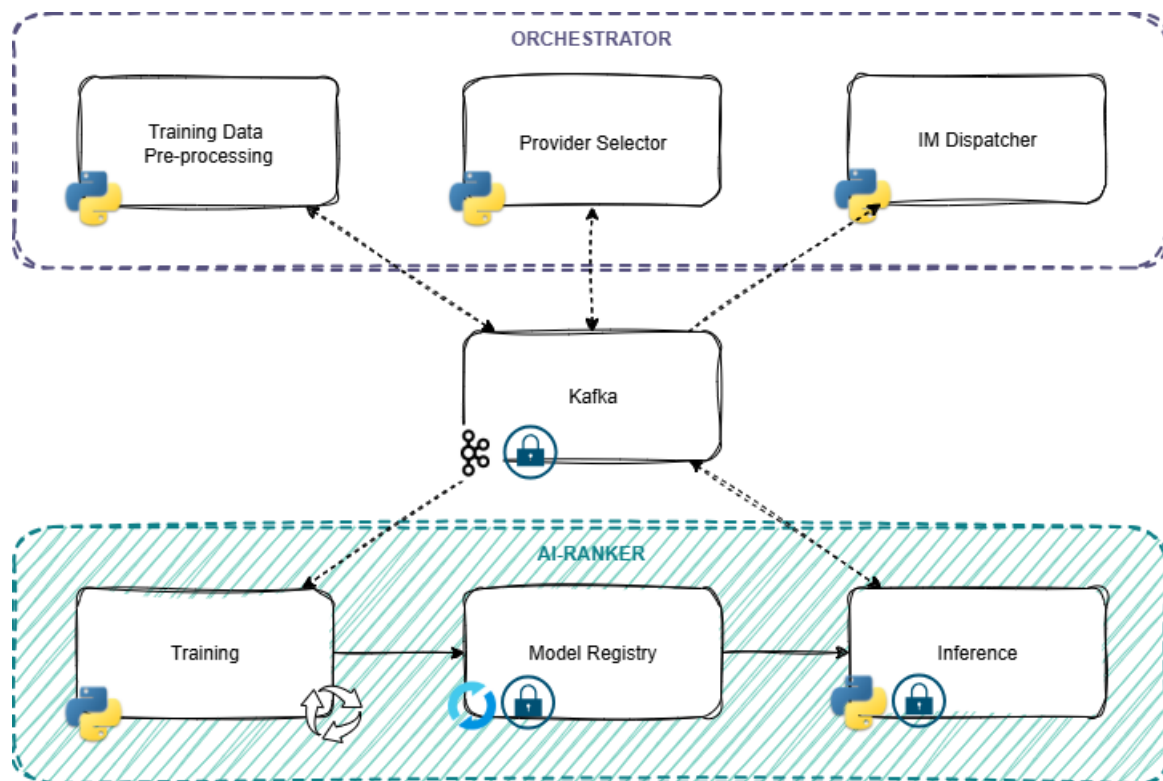


Architecture recap (1)



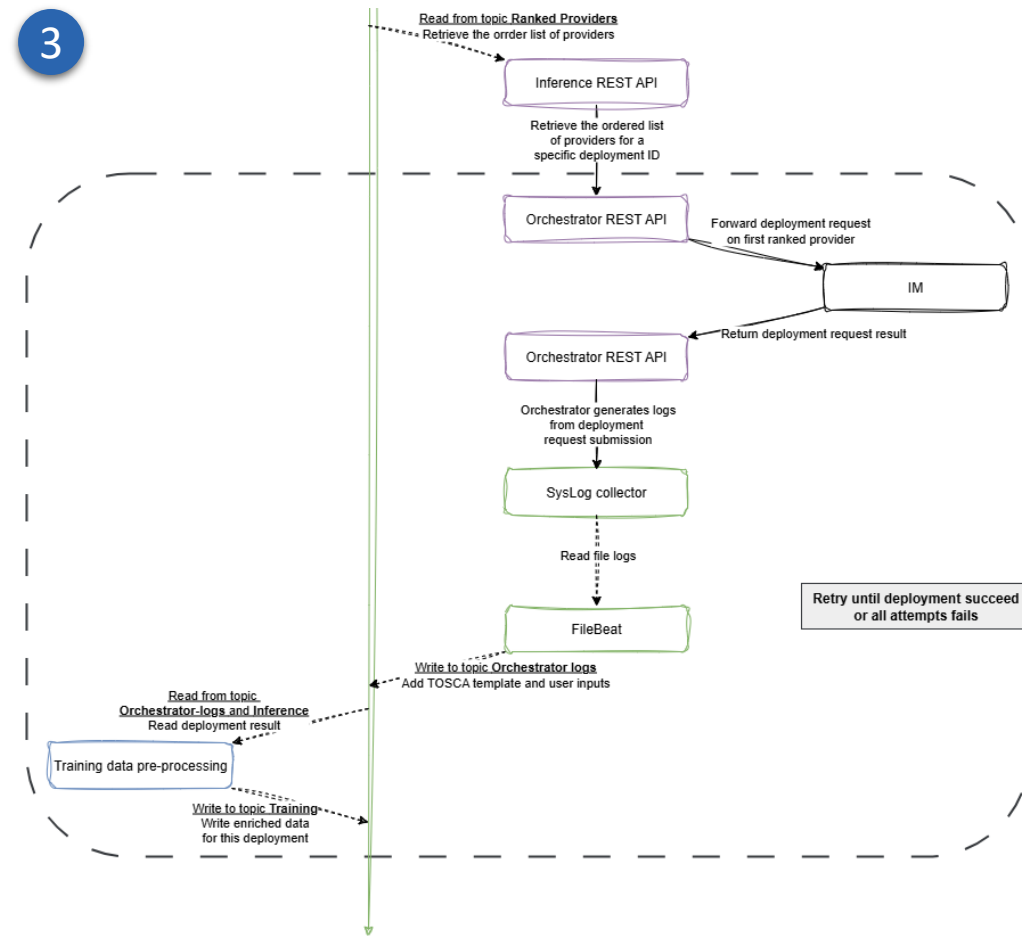
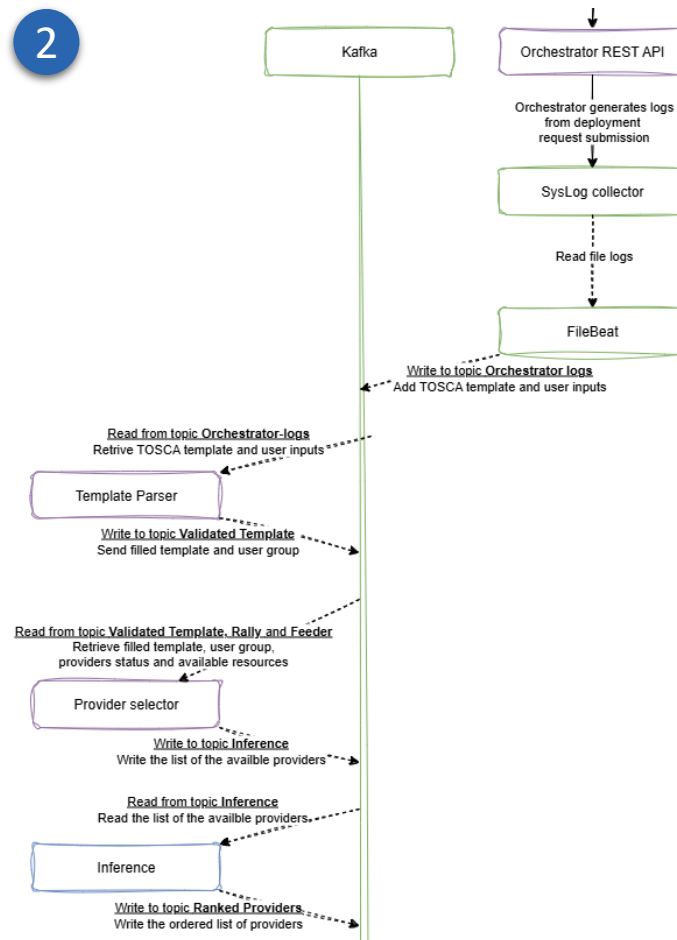
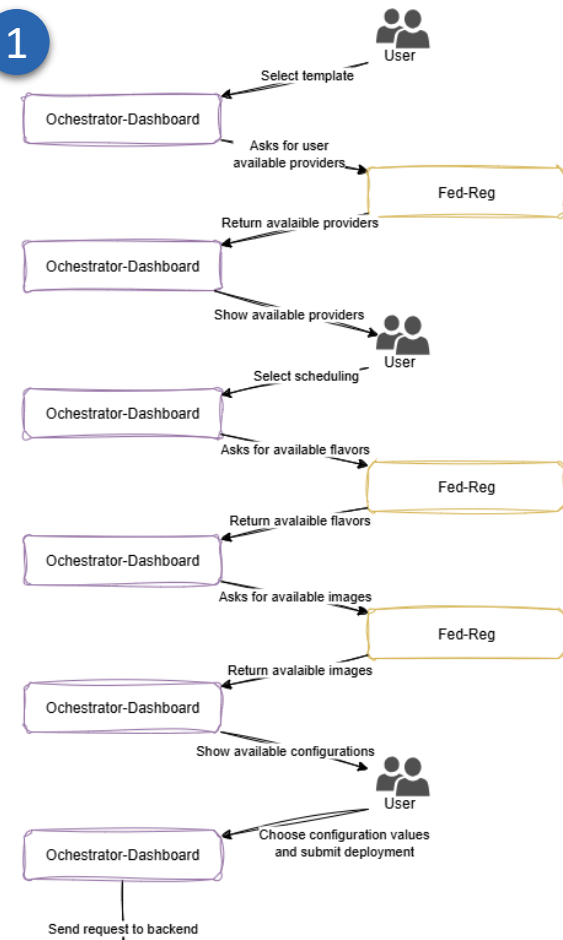


Architecture recap (2)



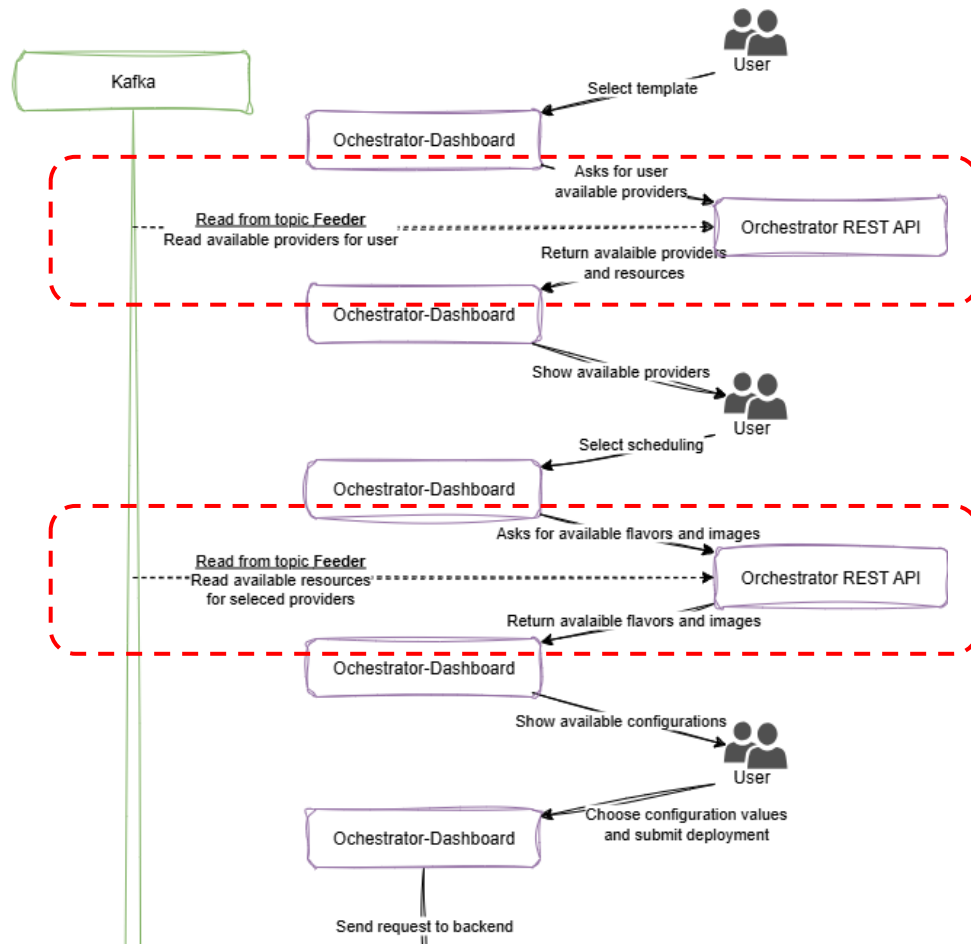


Deployment submission - State of the Art





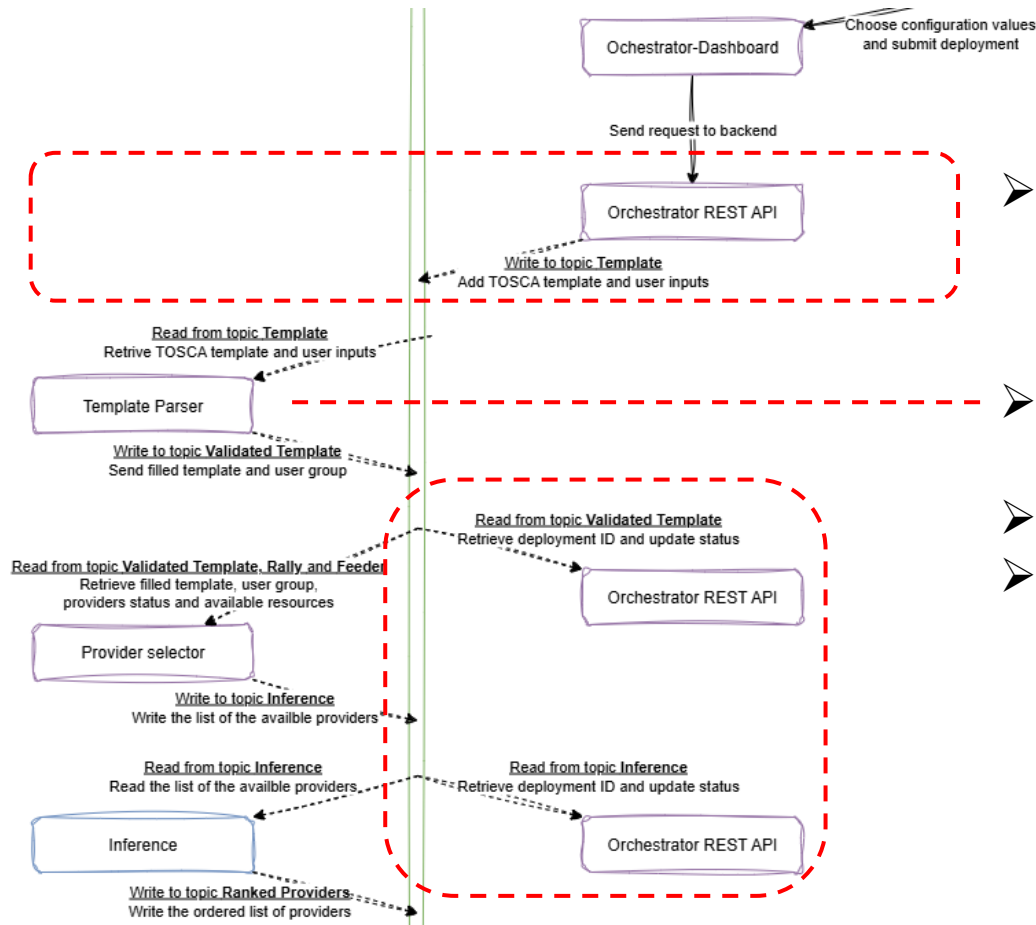
Deployment submission – Final version – Dashboard and REST API



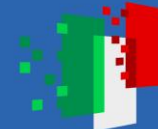
- The dashboard communicates only with the Orchestrator REST API, thus reduce calls to external services which may be not available
- The Orchestrator REST API reads only the last message of the topic
- Possibility to cache the response received from kafka (2nd call not needed)?



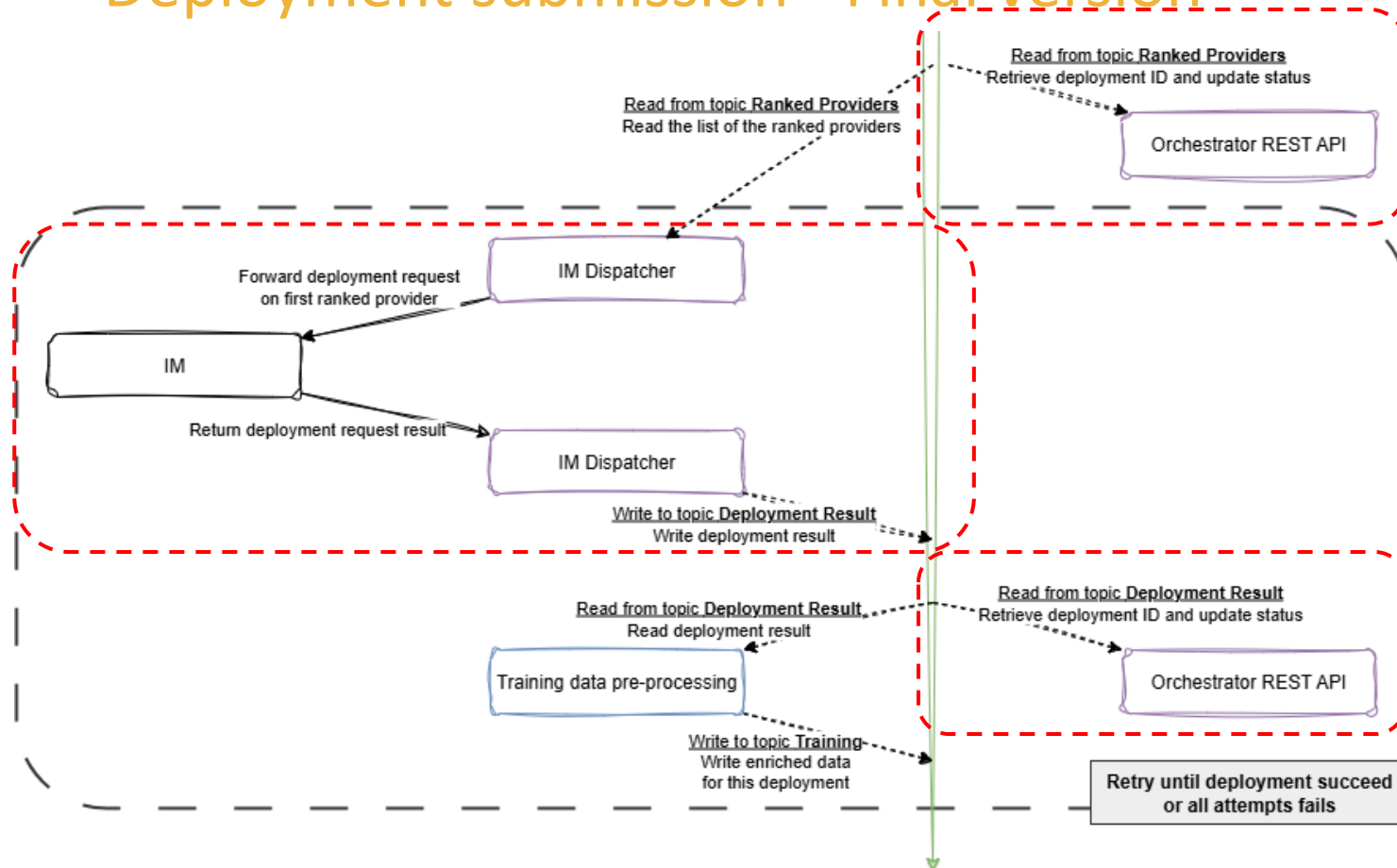
Deployment submission – Final version – REST API and Kafka



- Remove SysLog and corresponding component parsing logs
- Update to use the [openstack-tosca-parser](#)
- The REST API listen on all topics to track deployment status changes
- Alternatives:
 - Write a dedicated components with access to the REST API DB (really needed?)
 - All components writes on a dedicated topic «proc-status». The REST API reads only from this topic (similar to error management described in the following)



Deployment submission - Final version



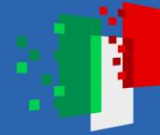
- The IM-Dispatcher manages the communication with the IM
- Manages both Kubernetes and Openstack deployments
- In case of failed deployments, retries until all ranked provider have been used or timeout or other configuration limits are reached
- For successful deployments, the REST API updates the resources related to that deployment
- For failed deployments, the REST API updates only the deployment status



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



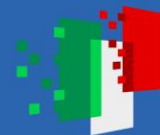
What if an error occurs in the procedure?

- The component raising the error must log the information and send a message on a dedicated kafka topic (to make it available to other components)
- The Orchestrator REST API reads the new message and updates the state of the corresponding deployment request.



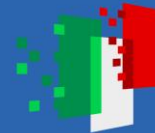
What if a component crashes unexpectedly?

- On startup, the components reads all the messages in the input and output topics.
 - Skip all messages having a corresponding value in the normal output topic or in the error one.
 - Resume and process only messages without a corresponding message in the output topics.
- Mechanism to detect if a component is dead
 - Use Kubernetes with ReplicaSet
 - Kubernetes and ArgoCD can notify about crashes?
 - It still does not start... We will think about this
- Each deployment has a timeout



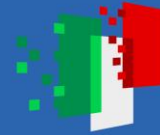
Kafka topics (1)

Topic name	Content description	Producers	Consumers
feeder	Federated provider configurations, resource quotas and available resources	- Feeder (periodically)	- Orchestrator REST API - Provider Selector
rally	Federated providers' status	- Rally monitoring	- Provider Selector
template	TOSCA template and user inputs	- Orchestrator REST API	- Template parser
Validated templates	TOSCA template and user inputs	- Template parser	- Orchestrator REST API - Provider Selector
inference	List of providers to rank and user inputs	- Provider Selector	- Orchestrator REST API - AI-Ranker inference service
Ranked providers	Ordered list of providers to use	- Inference service	- Orchestrator REST API - IM Dispatcher
Deployment results	Result of a deployment execution	- IM Dispatcher	- Orchestrator REST API - Training data pre-processing



Kafka topics (2)

Topic name	Content description	Producers	Consumers
training	Pre-processed data (deployment results and requested resources) for training	- Training data pre-processing	- AI-Ranker training script
errors	Generic procedure error	<ul style="list-style-type: none">- Orchestrator REST API- Template parser- Provider Selector- Inference Service- IM Dispatcher- Training data-preprocessing(?)- AI-Ranker training script(?)- Feeder(?)- Rally monitoring(?)	- Orchestrator REST API



Kafka topics (3)

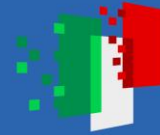
Topic name	Content description	Producers	Consumers
procedure-status (?) <div>Alternative to «Orchestrator REST API reads from all topics»</div>	Deployment procedure step	<ul style="list-style-type: none">- Orchestrator REST API- Template parser- Provider Selector- Inference Service- IM Dispatcher	<ul style="list-style-type: none">- Orchestrator REST API



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Thank you

giovanni.savarese@ba.infn.it,
luca.giommi@cnae.infn.it