TNO innovation for life

Douane Pilot 2024

Advice for the production environment of FEDeRATED

TNO Publiek › TNO 2024 R12234
20 november 2024

ICT, Strategy & Policy
www.tno.nl

TNO 2024 R12234 – 20 november 2024

Douane Pilot 2024

Advice for the production environment of FEDeRATED

Auteurs: Erwin Somers
Theodor Chirvasuta
Gert Kruithof

Rubricering rapport	TNO Publiek
Titel	TNO Publiek
Rapporttekst	TNO Publiek
Aantal pagina's	21 (excl. voor- en achterblad)
Aantal bijlagen	0
Opdrachtgever	Ministerie van Infrastructuur en Waterstaat
Projectnaam	Pilot Douane Singapore
Projectnummer	060.61355

Table of Contents

1 Summary

In the European FEDeRATED project (https://www.federatedplatforms.eu/), new concepts have been developed to enable data sharing in the logistics sector. An important use case is the Dutch Customs Authority (DCA), exchanging data with stakeholders such as Singapore and Portbase in an event driven architecture.

Until the end of 2024, the system “Pilot Douane Singapore” is deployed in the development environment of TNO. It is the intention of DCA, to increase its usage, increase the number of events exchanged, and the number of partners. Therefore, it is desirable that the ownership of the system and software is transferred to DCA and the software deployed in the production environment of DCA. This report provides an advice for the production environment, based on the use cases and requirements from the pilot and anticipating future growth.

An important aspect of the advice is the ownership and responsibility of the production environment. Until now, all software of the 3 parties (DCA, Portbase and Singapore) was deployed in one pilot environment. If the production environment would be configured in the same way, the DCA will effectively become a hosting provider for its partners. An important recommendation is to aim for the situation that each partner is responsible for the deployment and maintenance of its own software. This will require discussions with these partners to become effective. In order to be operational on January 1st 2025, the advice is to configure the production environment in separate clusters per partners so that the ownership and maintenance can be transferred seamlessly in a later stage.

Section 2 to 5 contain the scenarios, the use cases and the requirements. The analysis of requirements is limited to the requirements needed for the production environment. Hence, no requirements are listed on the software itself. Section 6 contains the advice to DCA on the production environment based on these requirements. Section 7 contains a deployment diagram showing parts of the advice more graphically.

2 Abbreviations

API	Application Programming Interface
DCA	Dutch Customs Authority
EDC	Eclipse Data Space Components
FEDeRATED	EU project for decentralized data collaboration in logistics
FR	Functional Requirement
HELM	Tool for managing Kubernetes packages
HTTPS	Hypertext Transfer Protocol Secure
JRE	Java Run time Environment
JSON-LD	Javascript Object Notation for Linked Data
NFR	Non-functional Requirement
RDF	Resource Description Framework
RML	RDF Mapping Language
SC	Scenario
SPARQL	Standard Protocol and RDF Query Language
TB	Terabyte
TRL	Technology Readiness Level
UC	Use Case
UUID	Universal Unique Identifier

3 Scenarios

Scenario	SC-001
Scenario Title	Portbase integration in the communication between Dutch Customs and Singapore
Description	Portbase receives on its back-end system, data from their customers and makes it available through a long polling endpoint. In the Portbase FEDeRATED node, a script runs continuously to listen for updates from Portbase. When a new logistics event is fetched from Portbase, it is converted to the FEDeRATED ontology and stored in the Portbase FEDeRATED Node. The Portbase FEDeRATED Node sends a message to NTP and DCA with a minimal event. If DCA or NTP is interested in more data, they send a message with a query back to the Portbase FEDeRATED Node. The Portbase FEDeRATED Node then sends a message with the full event. When an event arrives at the Singapore FEDeRATED node, a webhook is triggered which notifies and forwards the data to the Singapore back-end system.

Table 1: Typical scenario for the Pilot Douane Singapore

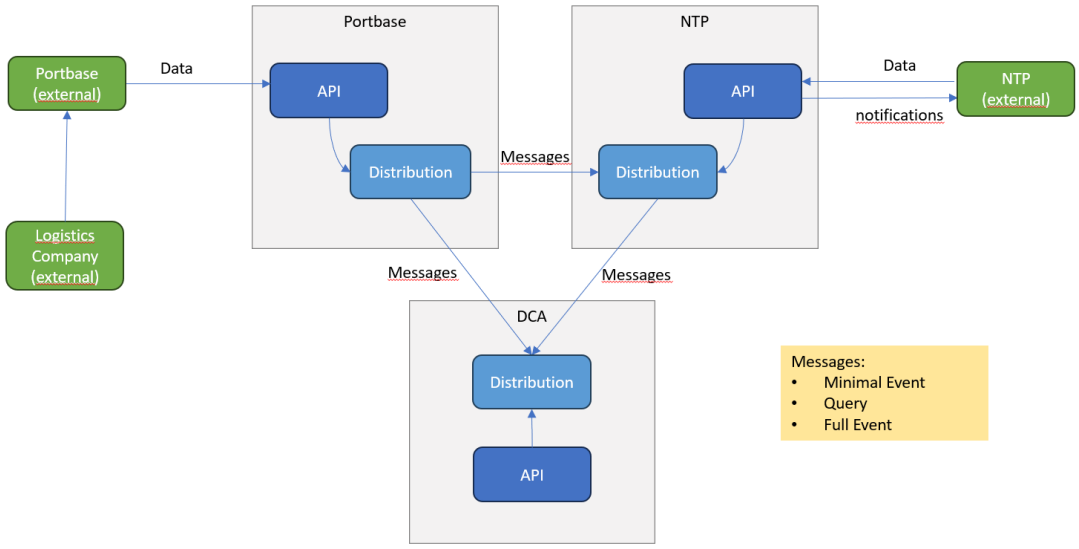

Figure 1: overview of Actors in Scenario 1. Grey boxes are the FEDeRATED Nodes.

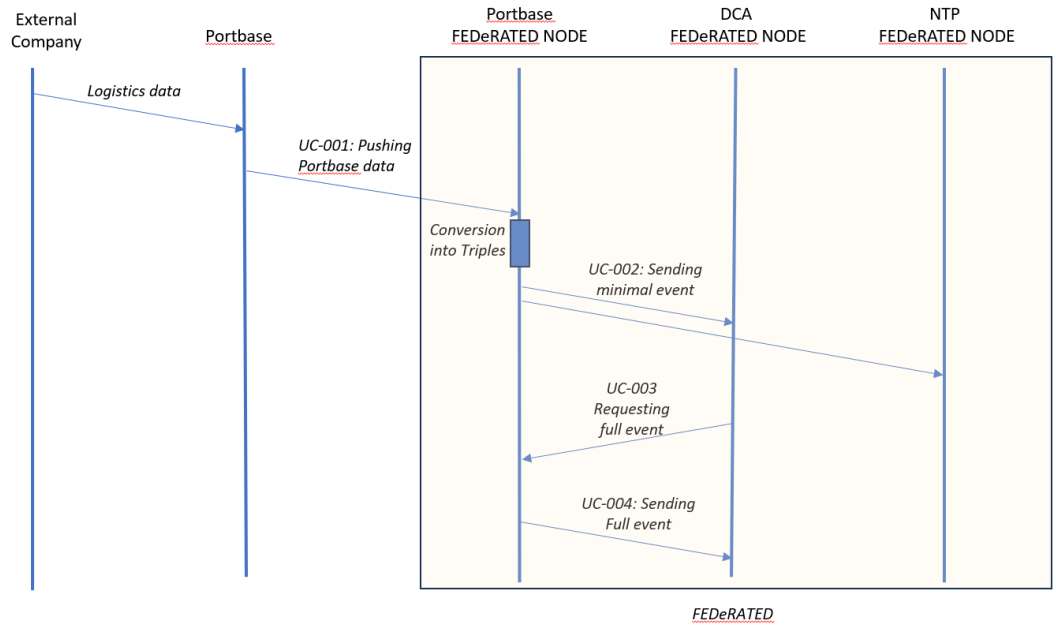

Figure 2: Swim lane view for a Scenario 1 (“happy flow” only)

4 Use Cases

4.1 Technical use cases

Use Case	UC-001
Related scenario's	SC-001
Use Case Title	Pushing Data
Description	Sending events to a FEDeRATED node
Actors	Initiating actor: Portbase; Destination actor: Portbase FEDeRATED Node
Precondition	Portbase FEDeRATED node installed Event-type configured
Steps	Portbase pushes full event data to its own FEDeRATED NODE. The data is converted into triples compliant with the FEDeRATED ontology. If successful, the Portbase FEDeRATED Node sends a signal back to Portbase and Portbase moves a pointer to avoid sending the same data again. If not successful, no message is sent and the pointer at Portbase is not moved. The step repeats until successful.
Post-condition	Events are ready for distribution.
Additional information	Volume: the expected number of events is circa 40.000 events/day. Availability information: • TRL 7, the availability should be >99%. • TRL 9: the availability should be >99.9% User interface information: on TRL7 the user interface is basic, intended for pilot execution. Information on number of users: on TRL7 the number of users in the system is < 5. Information on the number of participants: the system should be extendible to include stakeholders in Vietnam and in Germany.

Table 2: Use Case 1

Use Case	UC-002
Related scenario's	SC-001
Use Case Title	Sending Message with minimal Data
Description	The Portbase FEDeRATED Node sends a message that contains a minimal event to NTP and DCA
Actors	Initiating actor: Portbase FEDeRATED Node Destination actor: NTP FEDeRATED Node, DCA FEDeRATED Node
Precondition	• FEDeRATED nodes installed • FEDeRATED nodes reachable • Event-type configured • Data from Portbase converted to triples
Post-condition	Message is received at FEDeRATED Node of DCA and NTP
Additional information	None

Table 3: Use Case 2

Use Case	UC-003
Related scenario's	SC-001
Use Case Title	Requesting a Full Event
Description	The FEDeRATED Node of NTP or DCA sends a message to the FEDeRATED Node of Portbase to ask for the full event belonging to a minimal event.
Actors	Initiating actor: DCA FEDeRATED Node or NTP FEDeRATED Node Destination actor: Portbase FEDeRATED Node
Precondition	• FEDeRATED nodes installed • FEDeRATED nodes reachable • Event-types configured
Post-condition	Message with request of a Full Event is received at FEDeRATED Node of Portbase
Additional information	None

Table 4: Use Case 3

Use Case	UC-004
Related scenario's	SC-001
Use Case Title	Sending a Full Event
Description	The FEDeRATED Node of Portbase sends a message to the FEDeRATED Node of DCA or NCP to send the requested full event belonging to the associated minimal event.
Actors	Initiating actor: Portbase FEDeRATED Node Destination actor: DCA FEDeRATED Node or NTP FEDeRATED Node
Precondition	• FEDeRATED nodes installed • FEDeRATED nodes reachable • Event-types configured
Post-condition	Message with Full Event data is received at FEDeRATED Node of DCA or NTP
Additional information	None

Table 5: Use Case 4

4.2 Organizational use cases

Use Case	UC-005
Use Case Title	Deployment
Description	Deploying the FEDeRATED node components in a hosting environment
Actors	• Dutch Customs IT Department
Precondition	• Infrastructure available and reachable • Deployment tools available
Steps	• Setting the local parameters for the individual FEDeRATED nodes. • Setting up / configuring domain and DNS. • Run deployment script • Test FEDeRATED node (API and EDC) availability
Post-condition	FEDeRATED node API is reachable, EDC reachable, GraphDB is running and available
Additional information	None

Table 6: Use Case 5

User Case	UC-006
Use Case Title	Operations and Maintenance
Description	Managing and monitoring the correct operation of the FEDeRATED node and performing maintenance to ensure continuity and correct operations.
Actors	Dutch Customs IT Department.
Precondition	• FEDeRATED node is deployed and fully functional • An organizational process is in place • Knowledge and/or documentation on running and monitoring the FEDeRATED node id available
Steps	• A monitoring system is connected to the different FEDeRATED node parts to monitor availability • A backup method for the events is decided and scheduled • A restore procedure for the events is designed, documented and tested
Post-condition	The system is operational
Additional information	Retention period: the events of 3 months back should remain available in the backup. Further synchronization with back ends is out of scope.

Table 7: Use Case 6

5 Hosting requirements

5.1 System characteristics

FEDeRATED nodes are designed to be hosted as docker images.
A FEDeRATED node consists of the following images:

- The EDC connector
- The API (including event format verification and transformation)
- The event database (GraphDB)
- Distribution Orchestrator
- Postgresql database
- (depending on availability UIs for graphdb, edc, postgresql, events, etc.)

5.2 Continuity requirements

Functional Requirement	FR-001
Requirement Title	Continuity of a FEDeRATED Node
Related Use Cases	All
Requirement description	In case one component of the FEDeRATED node crashes, the system needs to make sure this component of the FEDeRATED node is restarted without depending on other FEDeRATED nodes
Priority	The continuity of a FEDeRATED node must not be dependent on the availability other FEDeRATED nodes
Verification method	Forcibly stop the FEDeRATED node (or part of it) , and check if the node (or part of it) is restarted.

Table 8: Functional Requirement 1

Functional Requirement	FR-002
Requirement Title	Continuity of monitoring
Related Use Cases	UC-005, UC-006
Rationale	In case a FEDeRATED Node stops working or crashes, the monitoring tools must keep working.
Requirement description	In case the FEDeRATED Node or the infrastructure, the FEDeRATED Node is running on, stops or crashes the monitoring tools must continue monitoring and report this unavailability of the FEDeRATED Node.
Priority	The monitoring and infrastructure tools must run independently from the FEDeRATED Nodes.
Verification method	Forcibly stop the infrastructure the FEDeRATED node is running

Table 8: Functional Requirement 2

5.3 Availability requirements

Non-Functional Requirement	NFR-001
Requirement Title	Availability at TRL 7
Related Use Cases	UC-001, UC-002, UC-003, UC-004
Requirement description	In order to receive / send events from / to a third party the FEDeRATED node must be available for the third party to connect to, in order to comply with TRL 7 and support processes within the organization.
Priority	The System must have an availability of 99% at TRL7
Verification method	Via monitoring tools

Table 10: Non-Functional Requirement 1

Non-Functional Requirement	NFR-002
Requirement Title	Availability at TRL 9
Related Use Cases and Requirement	UC-001, UC-002, UC-003, UC-004
Requirement description	In order to receive / send events from / to a third party the FEDeRATED node must be available for the third party to connect to, in order to comply with TRL 7 and support processes within the organization.
Priority	The System must have an availability of 99,9% at TRL9
Verification method	Via monitoring tools

Table 11: Non-Functional Requirement 2

5.4 Scalability requirements

Non-Functional Requirement	NFR-003
Requirement Title	Scalability for larger volumes of evets, more use cases and more parties
Related Use Cases and Requirement	UC-001, UC-002, UC-003, UC-004
Requirement description	More use cases are expected and more partners are expected to join.
Priority	The production environment must be scalable with the growth of the pilot
Verification method	Contractual

Table 12: Non-Functional Requirement 3

5.5 Access requirements

Functional Requirement	FR-003
Requirement Title	Outside access to a node
Related Use Cases	UC-001
Requirement description	The EDC module of the FEDeRATED Node must be able to receive events from and send events to other FEDeRATED Nodes
Priority	The production environment must have an ingress service to route outside access to the correct FEDeRATED Node.
Verification method	The ingress manager is deployed and operational.

Table 13: Functional Requirement 3

5.6 Monitoring requirements

Functional Requirement	FR-004
Requirement Title	Monitoring
Related Use Cases	UC-005, UC-006
Requirement description	Functionality and availability of the FEDeRATED node must be checked periodically to verify the correct functioning of the system
Priority	Monitoring tools and processes to verify availability and functionality must be in place
Verification method	Monitoring tools are available and connected to the FEDeRATED node and report availability.

Table 14: Functional Requirement 4

5.7 Deployment requirements

Functional Requirement	FR-005
Requirement Title	Deployment
Related Use Cases	UC-005, UC-006
Requirement description	Updates are implemented effortlessly and using standard deployment tool.
Priority	The system must be deployed using standard deployment tools that allow deployment updates
Verification method	Update testing in the test environment

Table 15: Functional Requirement 5

5.8 Security requirements

Functional Requirement	FR-006
Requirements Title	Access Control
Related Use Cases	UC-001, UC-002, UC-003, UC-004
Requirement description	Prevention of unauthorized access, misuse, and malicious attacks.
Requirement description	The data used and stored by the FEDeRATED node is sensitive and needs to be secure and out of reach of unauthorized parties.
Priority	The system must have a centralized point for managing security aspects.
Verification method	The API is secured with credentials and uses the required security and authorization methods. The API is only reachable using HTTPS. The deployed environment is only accessible using credentials.

Table 16: Functional Requirement 6

Functional Requirement	FR-007
Requirement Title	Integrity of the production environment
Related Use Cases	UC-001, UC-002, UC-003, UC-004
Requirement description	The integrity of the production environment is verified by a certificate, signed by a trusted authority, and used with HTTPS to enforce this certificate.
Priority	The production environment must use HTTPS protocol and certificates.
Verification method	Test

Table 17: Functional Requirement 7

Functional Requirement	FR-008
Requirement Title	Certificate management / process
Related Use Cases	UC-001, UC-002, UC-003, UC-004
Requirement description	The Certificates used for the HTTPS need to be valid in order to be trusted
Priority	The production environment must have a service or process to set, manage, and acquire certificates for the ingresses made to the deployed webservices.
Verification method	Test

Table 18: Functional Requirement 8

5.9 Software license requirements

Functional Requirement	FR-009
Requirements Title	Software Licenses
Related Use Cases	UC-005, UC-006
Requirement description	Because all products used in the FEDeRATED node are open source no Software licenses are needed.
Priority	The production environment won't have software that requires licenses.
Verification method	None

Table 19: Functional Requirement 9

5.10 Backup and Restore requirements

Functional Requirement	FR-010
Requirement Title	Back up service
Related Use Cases	UC-001, UC-003
Requirement description	A backup process and service will back up the stored configuration and event data. This backup can be restored in case of unexpected data loss on the FEDeRATION node. This way the continuity of the FEDeRATED node can be ensured
Priority	The production environment must have a service to create and restore backups of the event data.
Verification method	Test

Table 20: Functional Requirement 10

Non-Functional Requirement	NFR-004
Requirement Title	Backup capacity
Related Use Cases	UC-001, UC-003
Requirement description	The backup capacity is minimally equivalent to the amount of data the FEDeRATED nodes send and receives on a daily basis , multiplied by the amount of days of the data retention in the database. For UC-001 this contains the node settings and event types configured. For scenario UC-003 the event data is backed-up. The size is subjected to the amount of events and the definition of the event type
Priority	The production environment must have an additional capacity of 50 GB to store the back up of events.
Verification method	Test

Table 21: Non-Functional Requirement 4

6 Organisational requirements

Functional Requirement	FR-011
Requirement Title	Cloud provider policy
Related Use Cases	UC-005, UC-006
Requirement description	The DCA has a preference to use Azure as cloud platform
Priority	The production environment must be deployed in an Azure cloud environment
Verification method	Check organizational procedures

Table 22: Functional Requirement 11

Non-Functional Requirement	NFR-005
Requirement Title	Open Source Policy
Related Use Cases	UC-001, UC-002, UC-003, UC-004
Requirement description	The use of open source tools brings transparency to the internal workings of the software and vendor independency
Priority	The tools to be used should be Open Source when possible
Verification method	Check policies.

Table 23: Non-Functional Requirement 5

Non-Functional Requirement	NFR-006
Requirement Title	Organizational processes: Deployment and Maintenance
Related Use Cases	UC-005, UC-006
Requirement description	Deployment and maintenance of new software processes that are compatible with the existing processes in the organization
Priority	An IT process must be in place for deployment and maintenance of the FEDeRATED node
Verification method	Check organizational procedures

Table 24: Non-Functional Requirement 6

Non-Functional Requirement	NFR-007
Requirement Title	Organizational processes: Monitoring
Related Use Cases	UC-005, UC-006
Requirement description	Monitoring processes that are compatible with the existing processes in the organization
Priority	A process must be in place to monitor the availability of the FEDeRATED node by the IT department
Verification method	Check organizational procedures

Table 25: Non-Functional Requirement 7

Non-Functional Requirement	NFR-008
Requirement Title	Staging/Acceptation Environment
Related Use Cases	All
Requirement description	Testing changes and updates to the FEDeRATED node without risking the functionality and availability of the FEDeRATED node in production a development and test environment can be used to test updates and changes. Note: the ingress manager and the certificate manager can be shared between the acceptation environment and the production environment.
Priority	An Acceptance and staging possibility must be available for the roll out of upgrades of the FEDeRATED node
Verification method	

Table 26: Non-Functional Requirement 8

Non-functional Requirement	NFR-009
Requirement Title	Update and vulnerability monitoring
Related Use Cases	UC-005, UC-006
Requirement description	In order to keep up with updates and vulnerabilities, the organization has to be aware of them
Priority	The production environment should have a process to review and if needed implement the updates and changes to prevent vulnerabilities and ensure functionality of the FEDeRATED node.
Verification method	Test

Table 27 Non-Functional Requirement 9

Non-functional Requirement	NFR-010
Requirement Title	Hosting responsibility
Related Use Cases	All
Requirement description	Parties in the pilot should have full control of their nodes
Priority	Each FEDeRATED node must be deployed in a separate cluster under control of the respective party.
Verification method	Test

Table 28: Non-Functional Requirement 10

7 Advice

Nr	Recommendation	Related Requirements
R-001	It is recommended to run the images in a Kubernetes environment.	NFR-001, NFR-002, NFR-003, FR-011
R-002	It is recommended to host the Kubernetes using a cloud infrastructure. Hosting on local hardware will not comply with the combined non-functional requirements of availability, scalability and cost.	NFR-001, NFR-002, NFR-003, FR-011
R-003	It is recommended to use a Kubernetes cluster that can withhold docker images.	System Characteristics
R-004	It is recommended to deploy the software for each party in a separate Kubernetes Cluster. Alternatively the nodes of a set of organizations can be deployed in a single Kubernetes cluster. This is less costly in terms of monthly Microsoft Azure costs. However, DCA will then operate as a hosting provider for other organizations, which may not be desirable. It is possible to migrate from “one cluster for all nodes” to “one cluster per organization” in a later stage.	NFR-010
R-004	It is recommended to use the Microsoft Azure Platform.	NFR-001, FR-001, FR-002, FR-011
R-005	It is recommended to use Nginx ingress controller. It is an open source tool that manages incoming http and https call to the cluster and routes it through the internal Kubernetes infrastructure to the correct pod.	FR-003
R-006	It is recommended to use the “Backup Extension” of Azure to back up and restore and retain the data in the Azure resource group.	FR-010
R-007	It is recommended to implement the Azure Advisor recommendations.	FR-006, FR-007, FR-008
R-008	It is recommended to make use of a certificate manager and or process to keep the used certificates valid	FR-008
R-009	It is recommended to start the deployment with 16 GB of RAM and 1 TB of storage for each FEDeRATED Node.	System Characteristics, NFR-003, FR-010, NFR-004
R-010	It is recommended to use Helm scripts for deployment. Helm is the package manager for Kubernetes, that allowed for easy deployment of docker images into a Kubernetes cluster. Helm can update to newer versions of images, create ingresses, secrets and config maps in a single command using helm templates, specifically made for the deployment of the FEDeRATED node in Kubernetes. Also HELM keeps a history of releases in Kubernetes and supports sellback to previous versions.	FR-005

R-011	It is recommended to apply the customary DCA processes for technical and operational management of IT resources to the software. This should at least include: • Performing regular software updates, preferably based on active scanning of newer versions • Using separate environments for development, testing, acceptance testing and production (DTAP) • Active scanning of log files on all relevant levels (application level, Kubernetes level, ...) • Active scanning of security risks	NFR-006, NFR-007, NFR-008, NFR-009

Table 29: Summary of the Advice

8 Deployment Diagram

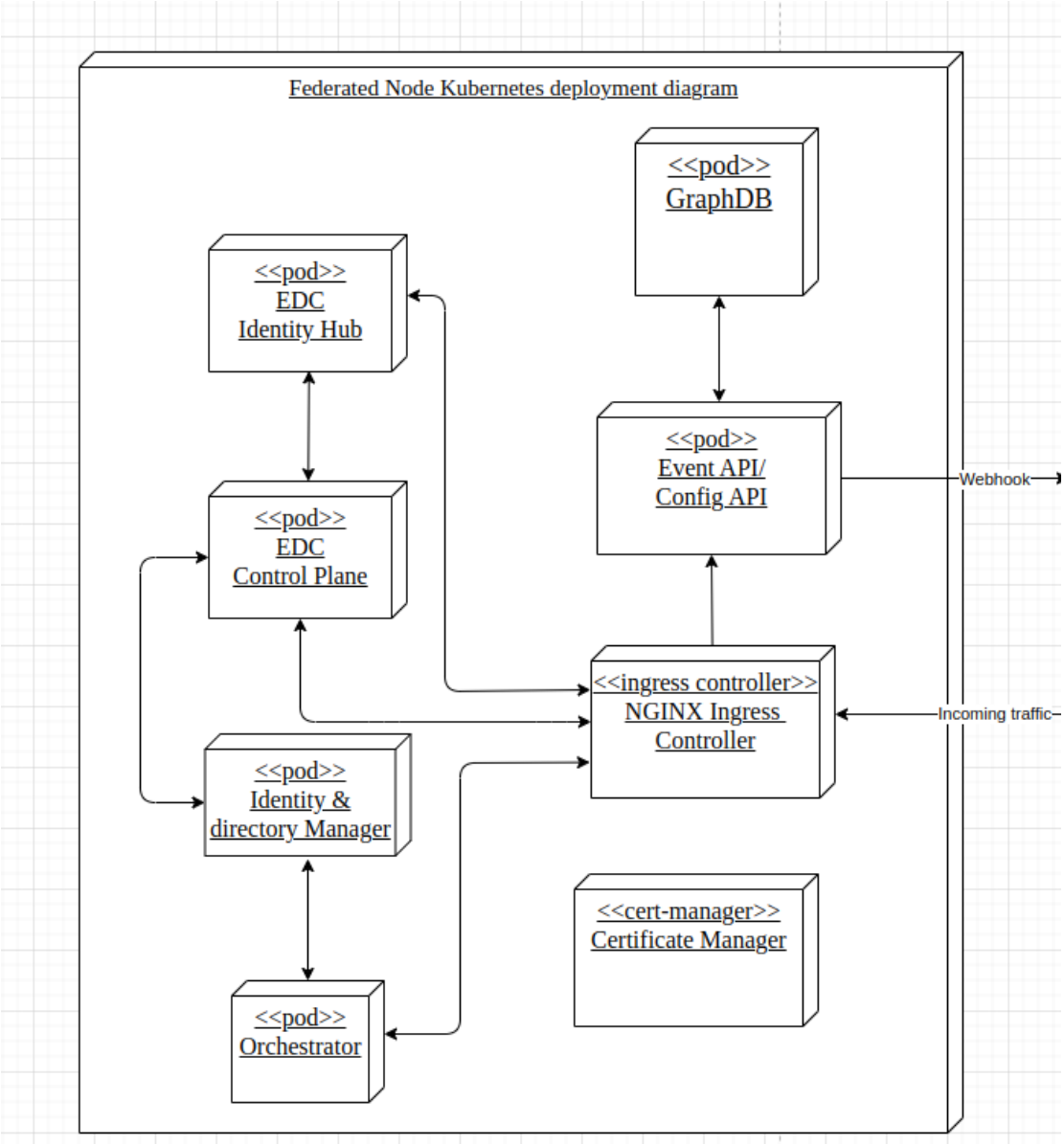

Figure 3: Deployment Diagram of FEDeRATED-EDC Node

ICT, Strategy & Policy

www.tno.nl

TNO innovation for life