

# PLEIADES

Smarter Plant Decommissioning



## PLEIADES DigiDecom Workshop

### Platform and tools integration presentation

Dusan Daniska

WAI

25 October 2023

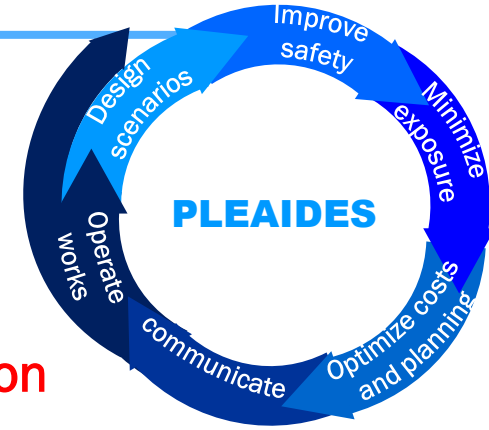


This project has received funding from the EURATOM Research & Training Programme 2014-2018 under the Grant Agreement n°899990. The content of this document reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.

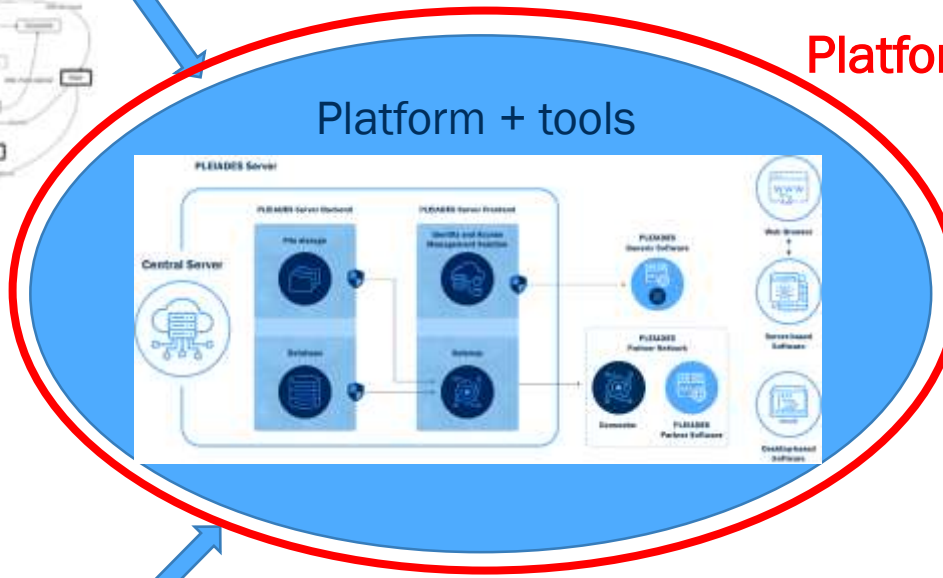
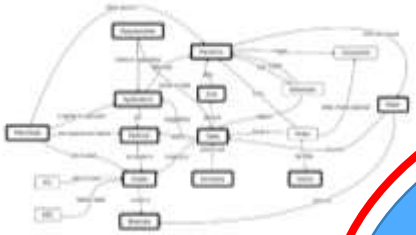
# Introduction

## PLEIADES BIM-based concept

BIM (*Building Information Modelling*) approach to **design scenario, improve safety, minimize radiation exposure, optimize costs and planning, communicate**



Data collection based on decommissioning ontology



## Platform and tools integration presentation

Scenario simulation

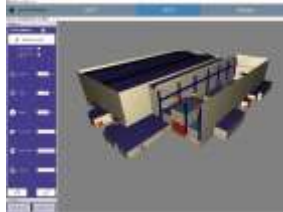
SMG



Estimations/Comparison/Decisions

- Waste management
- Dose exposure
- Cost
- Planning
- ...

3D models



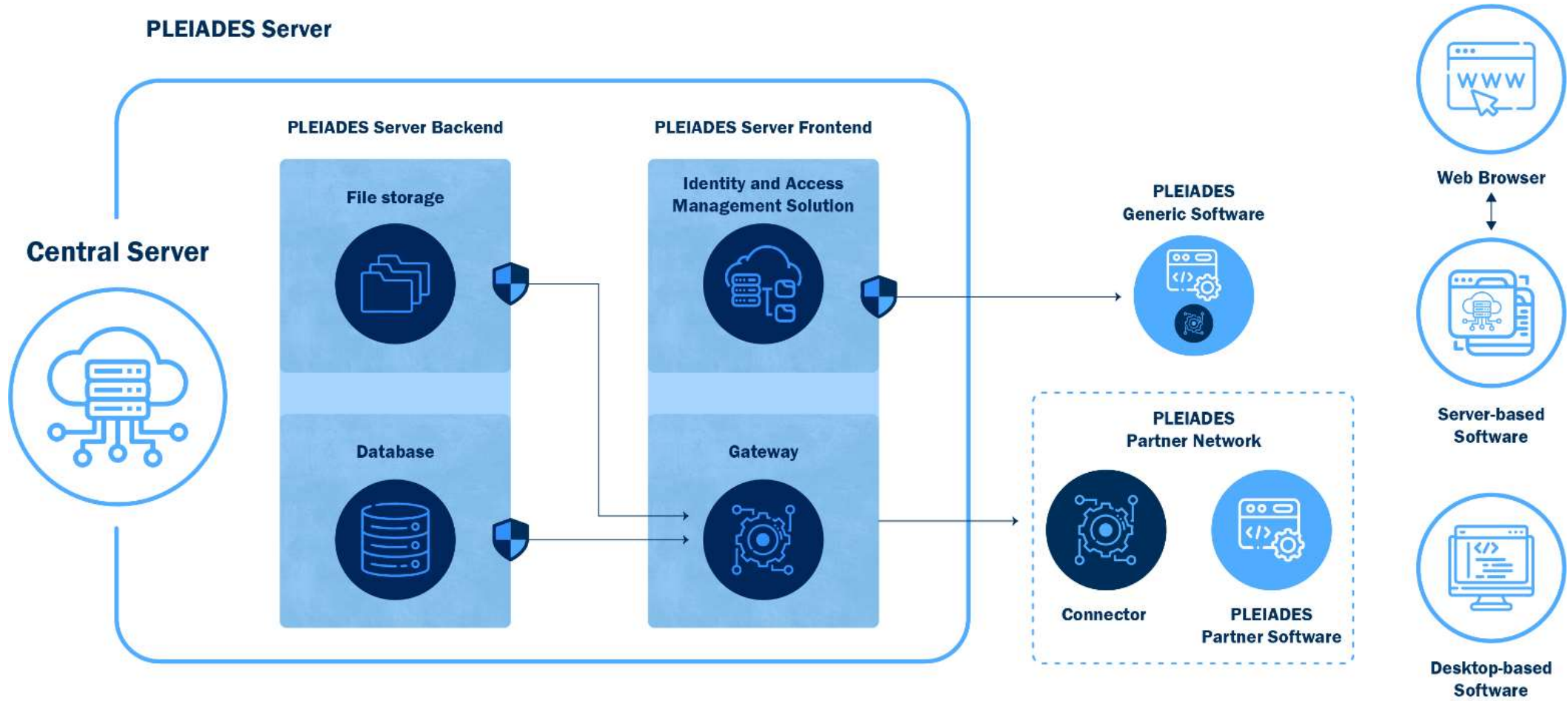
HRR



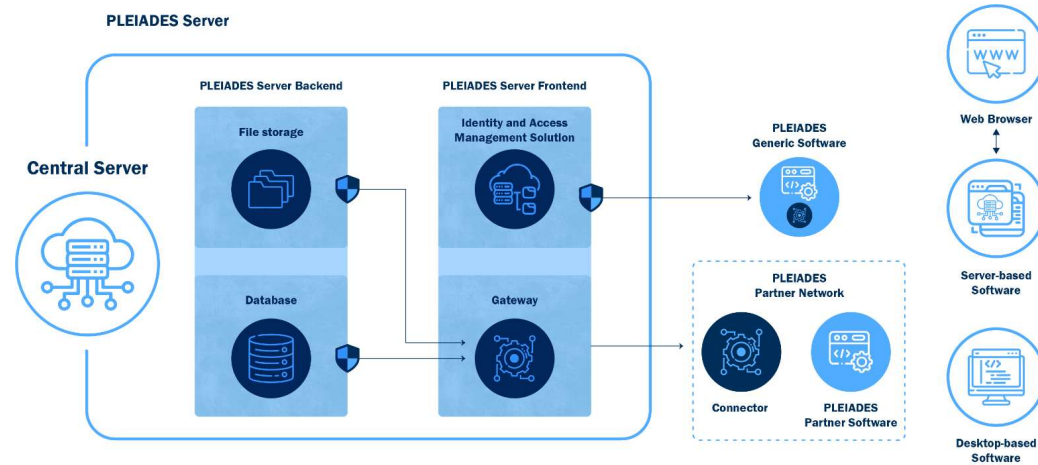
BCOT



# Architecture overview



# Software components



## Server side

- Authentication & authorization server (OAuth)
- Database engine
- File storage
- Database browser
- API classes browser
- API provider



## Client side

- PLEIADES connector (API consumer)



# PLEIADES API

|               |                          |               |  |
|---------------|--------------------------|---------------|--|
| <b>POST</b>   | /database/{databaseName} | <b>POST</b>   | /database/{databaseName}/record            |
| <b>GET</b>    | /databases               | <b>PUT</b>    | /database/{databaseName}/record/{recordId} |
| <b>DELETE</b> | /database/{databaseName} | <b>DELETE</b> | /database/{databaseName}/record/{recordId} |

- Authenticated using OAuth protocol
- Many databases can be managed
- Uses flexible JSON structures
- Structured data and files can be processed
- Records are classified based on the ontology
- Well described and documented
- **Easy to implement**

👉 The API specification and sample implementation of the client are publicly available.

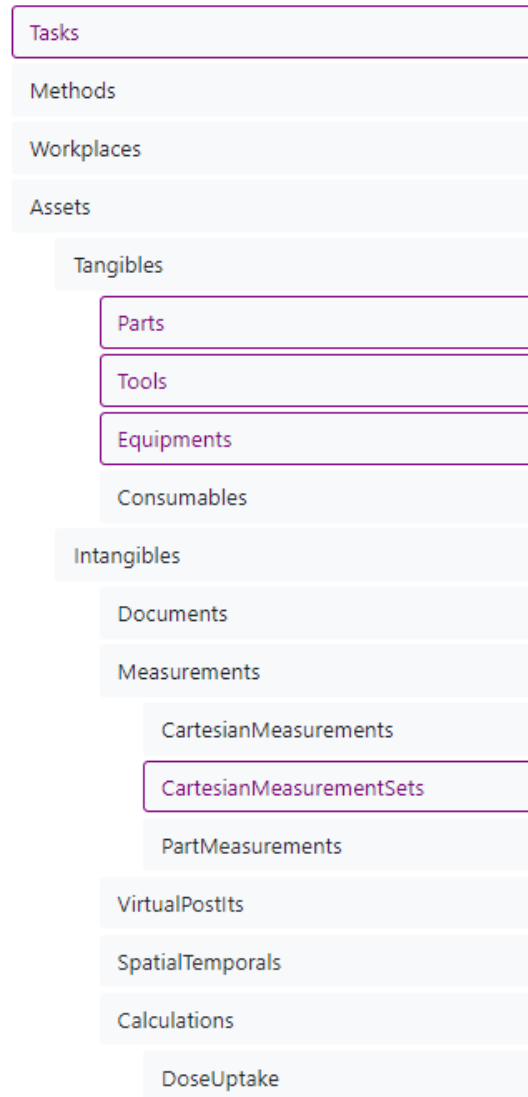
| Name   | Description   |
|--|---|
| <b>authorization</b> * required<br>(header)        | Example : Bearer MY_ACCESS_TOKEN<br><input type="text" value="Bearer MY_ACCESS_TOKEN"/>   |
| <b>databaseName</b> * required<br>string<br>(path) | Name of the database where the record will be stored.<br><input type="text" value="databaseName"/>  |
| <b>class</b> * required<br>string<br>(body)        | Class to which the record is categorized.<br>Example Value   Schema<br><input type="text" value="Assets.Tangibles.Parts"/><br>Parameter content type<br><input type="text" value="application/json"/>   |
| <b>content</b> * required<br>object<br>(body)      | Content of the new record. Metadata about creation datetime etc. will be added automatically.<br>Example Value   Schema<br><pre>{   "Name": "Cutting pipe",   "Number": "Task.1234",   "Start": "2022-03-28T09:37:51",   "Duration": 18,   "TeamId": "Hs1FTT2wXj91DxSx5",   "ApplicationId": "Hs1FTT2wXj91DxSx5",   "WorkplaceId": "Hs1FTT2wXj91DxSx5",   "MethodIds": [     "1w_Hs1FTT2wXj91DxSx5",     "3A5j_91DxSHs1FTT2wXj91DxSx5"   ] }</pre><br>Parameter content type<br><input type="text" value="application/json"/> |



# API classes (based on the ontology)

- Each record in database is classified
- Classes are structured based on the ontology
- Each class defines the structure of data

👍 The specification of API classes is publicly available.

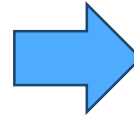
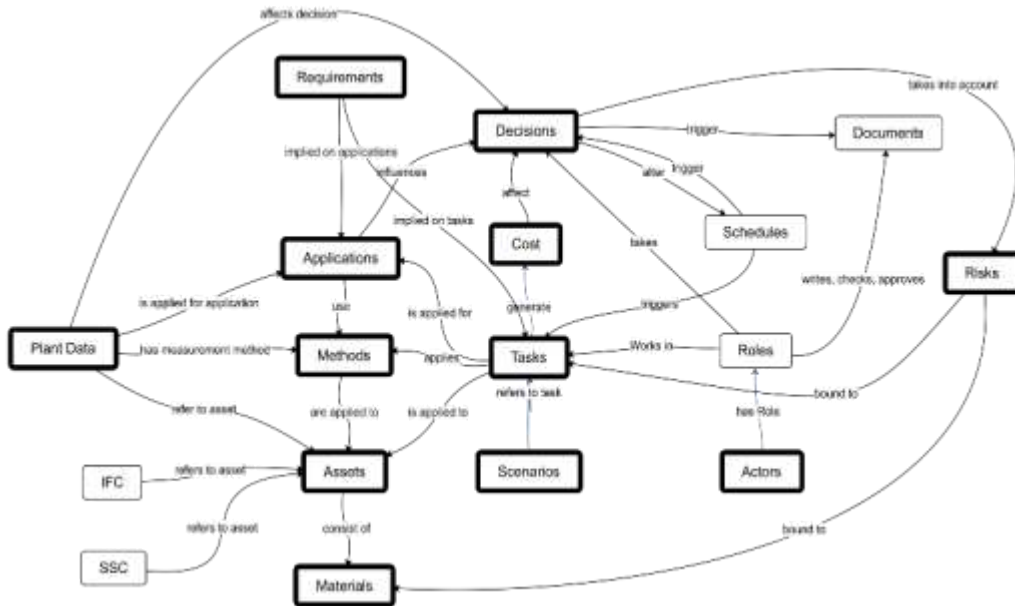


Example value of *body* in POST /record

```
{
  "class": "Tasks",
  "content": {
    "_id": "f8bf5894-f909-165d-5a68-3eae1d7cde8",
    "RecordInfo": {
      "CreatedOn": "2023-09-13T08:21:59",
      "CreatedBy": "any-string-value-random-6686",
      "ModifiedOn": "2023-09-13T08:21:59",
      "ModifiedBy": "any-string-value-random-8898"
    },
    "Name": "any-string-value-random-7574",
    "Number": "any-string-value-random-7772",
    "Definition": "any-string-value-random-2637",
    "ParentTaskId": "3f9b6058-c8ca-39ad-f4b1-7e277e2380d2",
    "DependsOnTaskIds": [
      "edd10f62-d641-567b-921e-5f064fdf686a",
      "3eb0c0da-c2e5-045e-14fd-babe740be085",
      "e845cc2c-6f7d-f010-95d3-f5309a47226f",
      "4b10a813-a818-4a4e-451d-95723ed5aa2e",
      "8a4cf634-ee41-06cb-1d46-f930d29ed6f6"
    ],
    "Start": "2023-09-13T08:21:59",
    "Duration": 877.508,
    "EstimatedDuration": 783.514,
    "TeamId": "150130c0-e8b7-f8c4-5111-1846f1ce5056",
    "ApplicationId": "df730cc0-c5eb-ee3d-bf38-85664a9d1741",
    "WorkplaceId": "561f7082-057b-29c8-372f-e1bf7405610a",
    "InputAssetIds": [
      {
        "Class": "Assets.Tangibles.Parts",
        "RecordId": "2ba907f6-289e-c411-7403-20623ee0584a"
      },
      {
        "Class": "Assets.Tangibles.Tools",
        "RecordId": "8f65e3ea-d453-ccfd-9acf-e4c940cc1cf0"
      },
      {
        "Class": "Assets.Tangibles.Consumables",
        "RecordId": "8f65e3ea-d453-ccfd-9acf-e4c940cc1cf0"
      }
    ]
  }
}
```



# API classes (based on the ontology)



- Tasks
- Methods
- Workplaces
- Assets
- Tangibles
  - Parts
  - Tools
  - Equipments
  - Consumables
- Intangibles
  - Documents
  - Measurements
    - CartesianMeasurements
    - CartesianMeasurementSets
    - PartMeasurements
  - VirtualPostIts
  - SpatialTemporals
  - Calculations
    - DoseUptake

Example value of body in POST /record

```
{
  "class": "Tasks",
  "content": {
    "_id": "f8bf5894-f909-165d-5a68-3eae1d7cde8",
    "RecordInfo": {
      "CreatedOn": "2023-09-13T08:21:59",
      "CreatedBy": "any-string-value-random-6686",
      "ModifiedOn": "2023-09-13T08:21:59",
      "ModifiedBy": "any-string-value-random-8898"
    },
    "Name": "any-string-value-random-7574",
    "Number": "any-string-value-random-7772",
    "Definition": "any-string-value-random-2637",
    "ParentTaskId": "3f9b6058-c8ca-39ad-f4b1-7e277e2380d2",
    "DependsOnTaskIds": [
      "edd10f62-d641-567b-921e-5f064fdf686a",
      "3eb0c0da-c2e5-045e-14fd-babe740be085",
      "e845cc2c-6f7d-f010-95d3-f5309a47226f",
      "4b10a813-a818-4a4e-451d-95723ed5aa2e",
      "8a4cf634-ee41-06cb-1d46-f930d29ed6f6"
    ],
    "Start": "2023-09-13T08:21:59",
    "Duration": 877.508,
    "EstimatedDuration": 783.514,
    "TeamId": "150130c0-e8b7-f8c4-5111-1846f1ce5056",
    "ApplicationId": "df730cc0-c5eb-ee3d-bf38-85664a9d1741",
    "WorkplaceId": "561f7082-057b-29c8-372f-e1bf7405610a",
    "InputAssetIds": [
      {
        "Class": "Assets.Tangibles.Parts",
        "RecordId": "2ba907f6-289e-c411-7403-20623ee0584a"
      },
      {
        "Class": "Assets.Tangibles.Tools",
        "RecordId": "8f65e3ea-d453-ccfd-9acf-e4c940cc1cf0"
      }
    ],
    {
      "Class": "Assets.Tangibles.Consumables",
      "RecordId": "8f65e3ea-d453-ccfd-9acf-e4c940cc1cf0"
    }
  ]
}
```



# Roadmap

---

- **Continue in the development** of the ontology and the API
- **Extend the coverage** of the decommissioning planning activities that can be supported by the PLEIADES platform
- **Utilization of other digitalization technologies** like artificial intelligence, robotics or integration with sensor networks





# Conclusion

1. The PLEIADES platform is an **open, robust and flexible platform** for digitalization of decommissioning planning.

**Open** = the specification of API is publicly available

**Robust** = large amount of data can be processed (3D models, point clouds, structured data)

**Flexible** = any software, independent from the technology, can connect and benefit from the common data environment

Publicly available results:

1. Specification of the API
2. Specification of the API classes
3. Sample implementation of the client (the “PLEIADES connector”)

2. The functionality of the PLEIADES platform has been proven by several software tools.

