# SpaceCube GmbH

## End-to-End System Engineering Portal (ESEP)

## Final Presentation

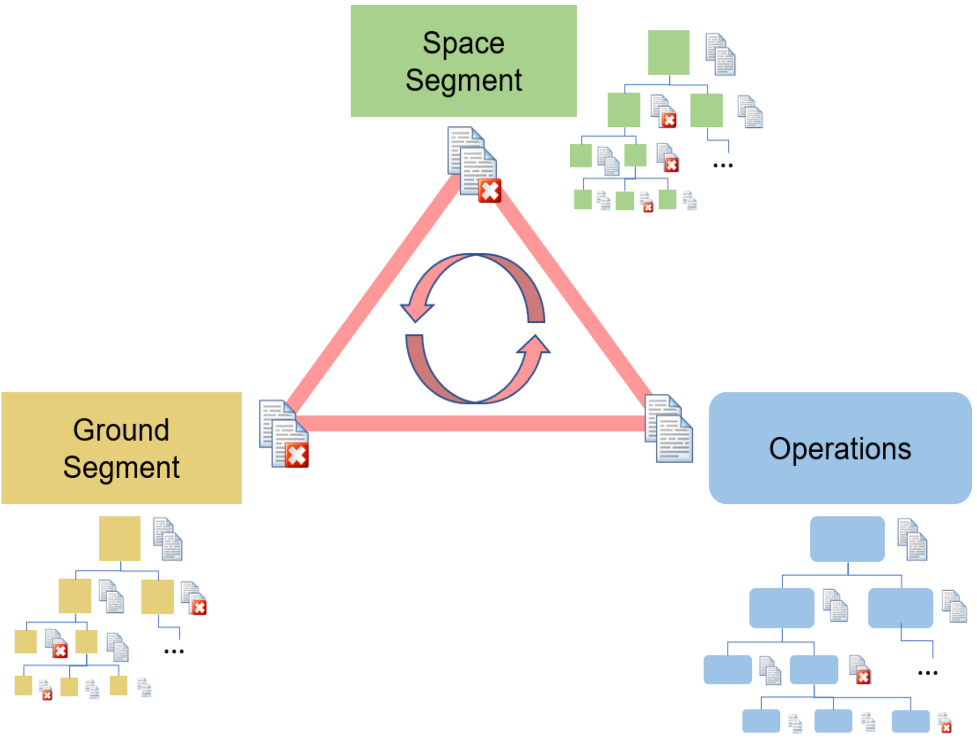
*Dr. Todor Stoitsev*

# Agenda

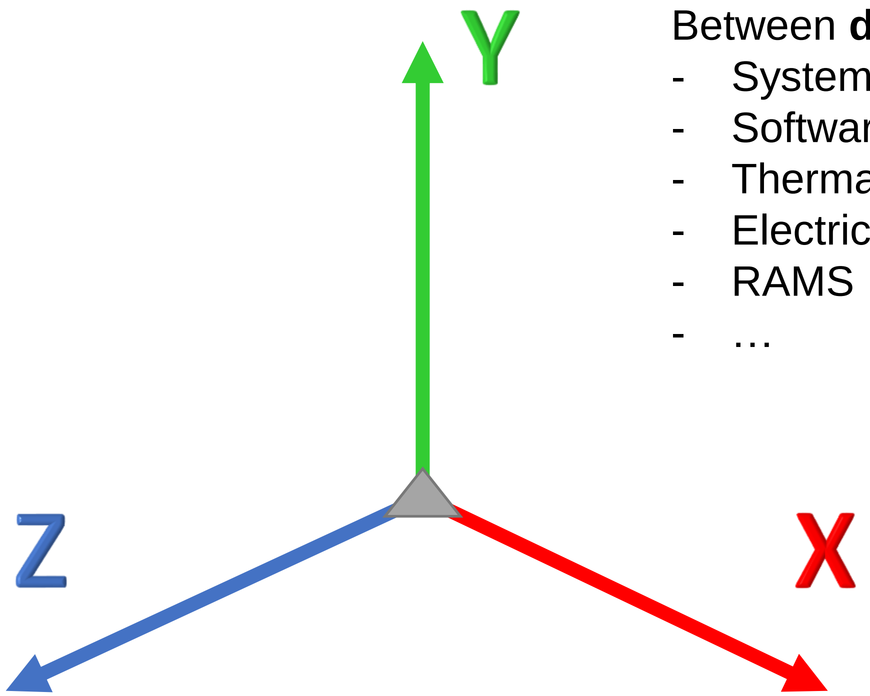
1. Introduction and objectives
2. Work logic and deliverables
3. Project team
4. Schedule
5. System concept
6. Main achievements
7. Design and implementation
8. Evaluation
9. Conclusions
10. Next steps

# 1. Introduction and objectives – space systems engineering

- **Space Systems Engineering:** complex + document-centric = costly and error prone
- **Model-Based Systems Engineering (MBSE):** "the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" – reduce engineering risks and costs
- **Model-Based for System Engineering (MB4SE) initiative** addressing the MBSE adoption across ESA, national space agencies, and Large System Integrators (LSIs), featuring:
  - **Common Space System Ontology (SSO)** - establishing common semantics in line with ECSS and ensuring semantic interoperability for the engineering data
  - **Model Based Engineering Hub (MBEH)** – providing a common infrastructure for engineering data exchange in line with the semantics established through the common space systems ontology
- **Amendment of ECSS processes and DRDs** towards model-centric deliverables as much as: 80% of ECSS-E-ST-10 standards and between 35-60% for ECSS-E-ST-70 standards and ECSS-E-ST-40 foreseen by MB4SE



# 1. Introduction and objectives – the problem of interoperability




Between **disciplines**
- Systems engineering
- Software
- Thermal
- Electrical
- RAMS
- …

Throughout the **project life cycle**
- Early phases
- Definition
- Qualification and production
- Utilization
- Disposal

Along the **supply chain**
- Customer
- Prime
- Different levels of subcontractors

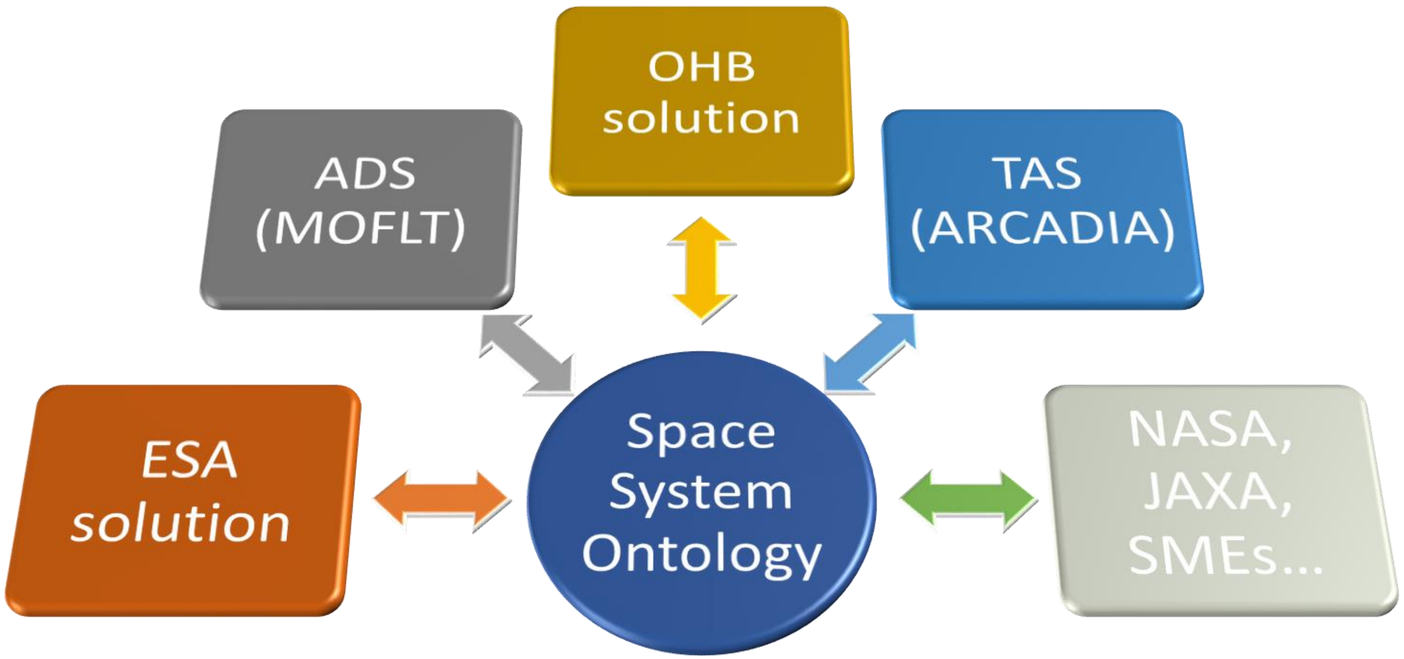
Similar semantics

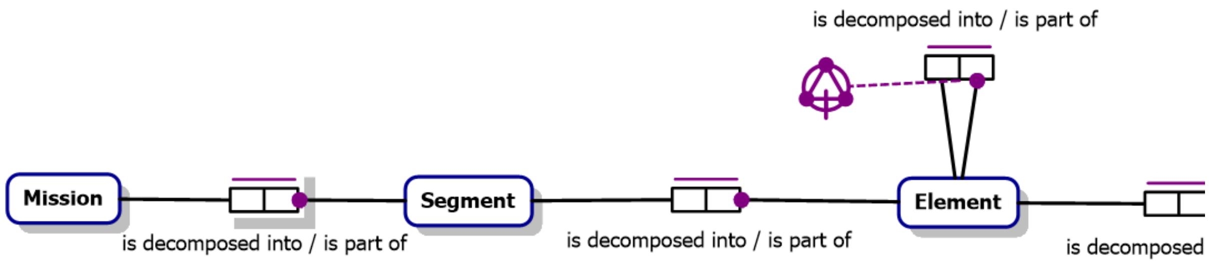
Different data models

Different data formats

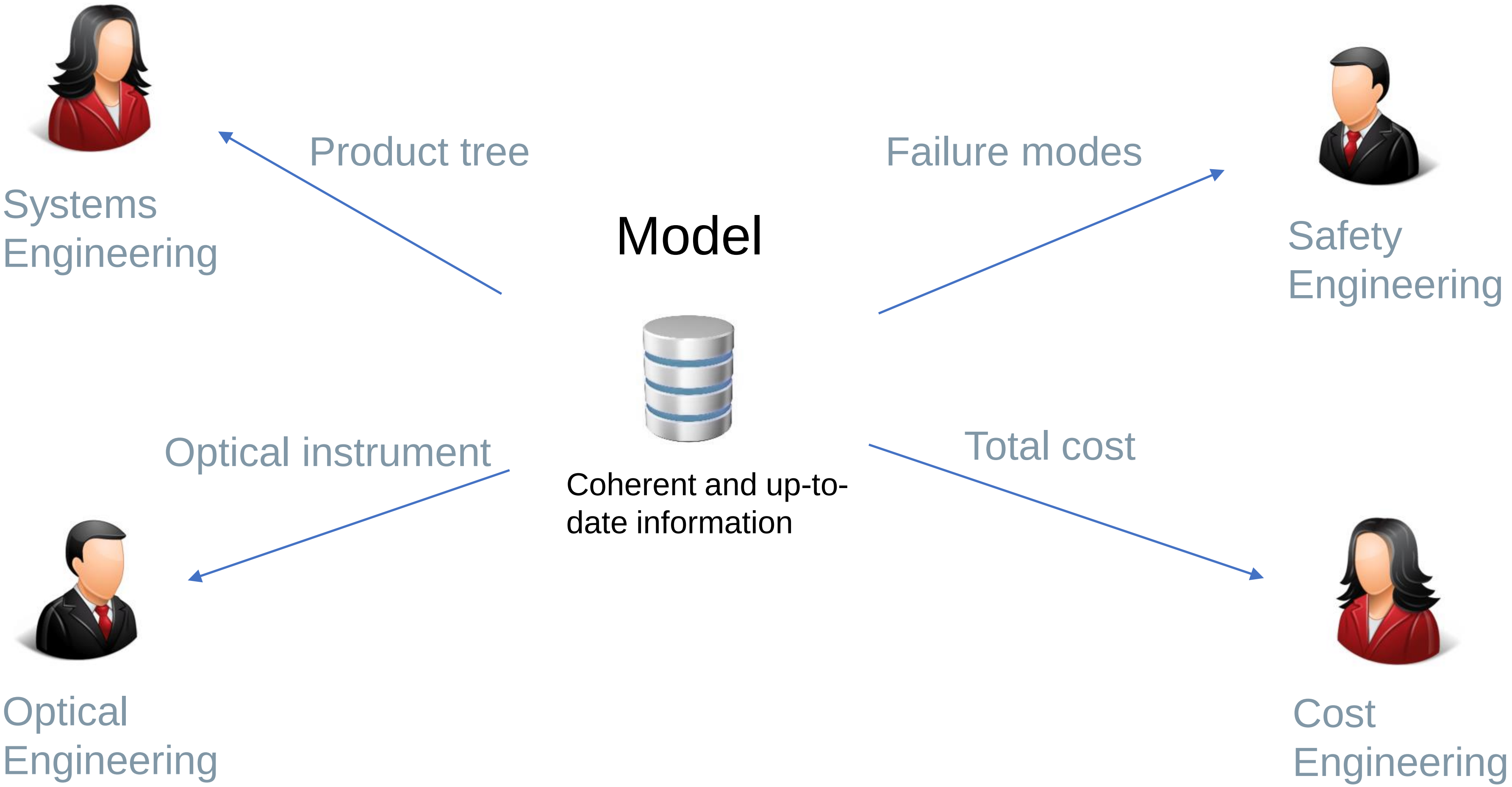
Different tools

# 1. Introduction and objectives – OSMoSE's ontology

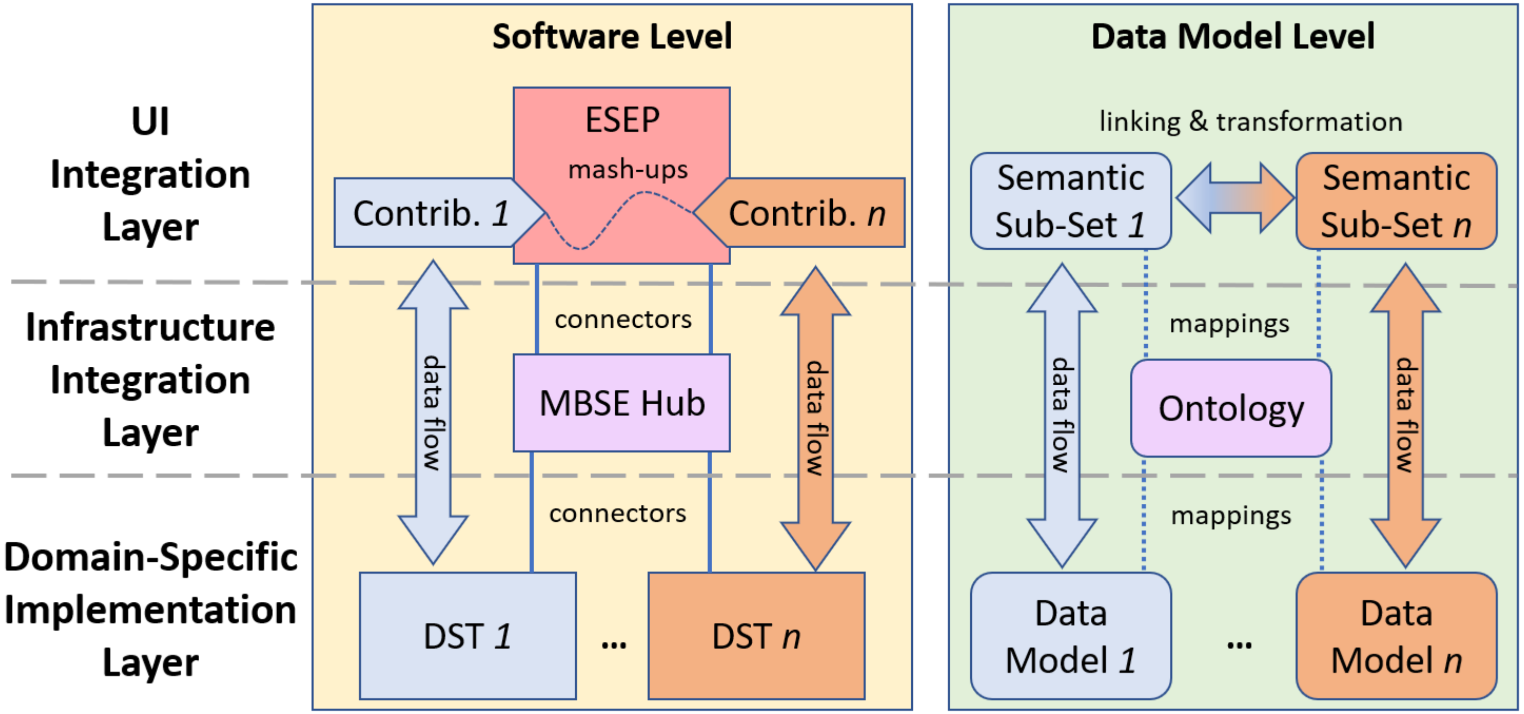


# 1. Introduction and objectives – Global System Model



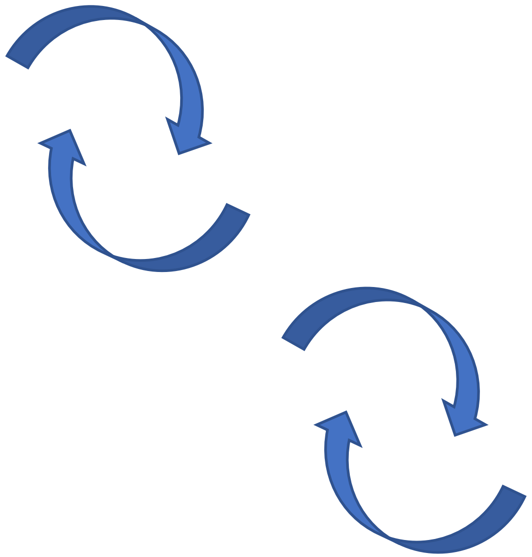
# 1. Introduction and objectives




- **Elaborate and demonstrate at proof-of-concept level a web-based, federated User Interface (UI) layer** supporting the engineering data integration and transformation across the space systems engineering lifecycle in line with the Model Based Engineering Hub (MBEH) concept and supporting the common Space System Ontology (SSO) semantics
- **Lower the barriers for the adoption of the overall MBSE approach for space systems** by providing intuitive UI techniques for user interaction with formal engineering data

# 2. Work logic and deliverables

- ❑ **WP1000 Analysis**
  - ❑ WP1100 Use cases and requirements analysis
  - ❑ WP1200 State-of-the-art analysis
  - ➔ **SRR:** Technical Note (TN) - User Requirements
- ❑ **WP2000 Design and development**
  - ➔ **PDR:** Software Specification and Design Document (SSDD)
- ❑ **WP3000 Demonstration**
  - ➔ **QR:**
    - Software User Manual (SUM)
    - Evaluation Report
    - Final Report
    - Executive Summary Report
    - Final Presentation slides
    - Illustration of the activity in one self-standing image
    - Ca. 5 min video summarizing the main results of the activity
    - ESEP prototype software



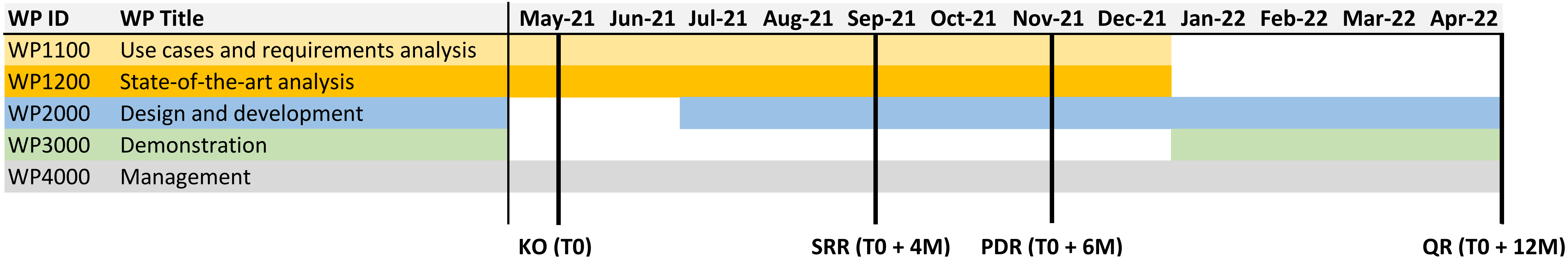
# 3. Project team

- **SpaceCube – prime / ESEP engineering team**

- **OHB – providing system engineering consultancy supporting:**
  - WP1000 Use Cases and User Requirements – contributions to Technical Note
  - WP3000 Demonstration – independent (from development team) evaluation through manual execution of test cases covering the main ESEP prototype features by representative system engineering end users – contribution to Evaluation Report

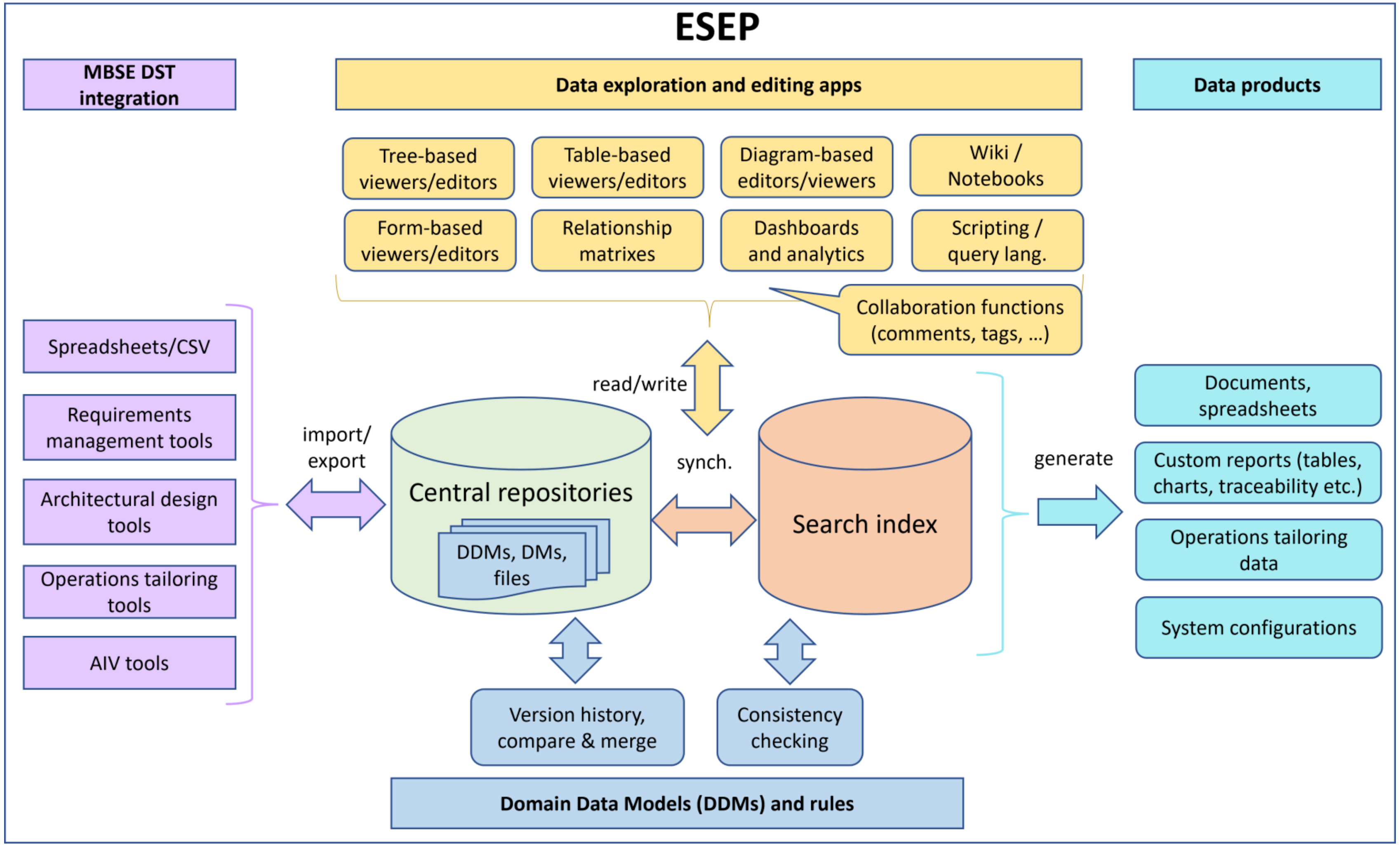
# 4. Schedule

- Total duration of ~12 months
- Agile approach with early prototyping for de-risking major technology areas



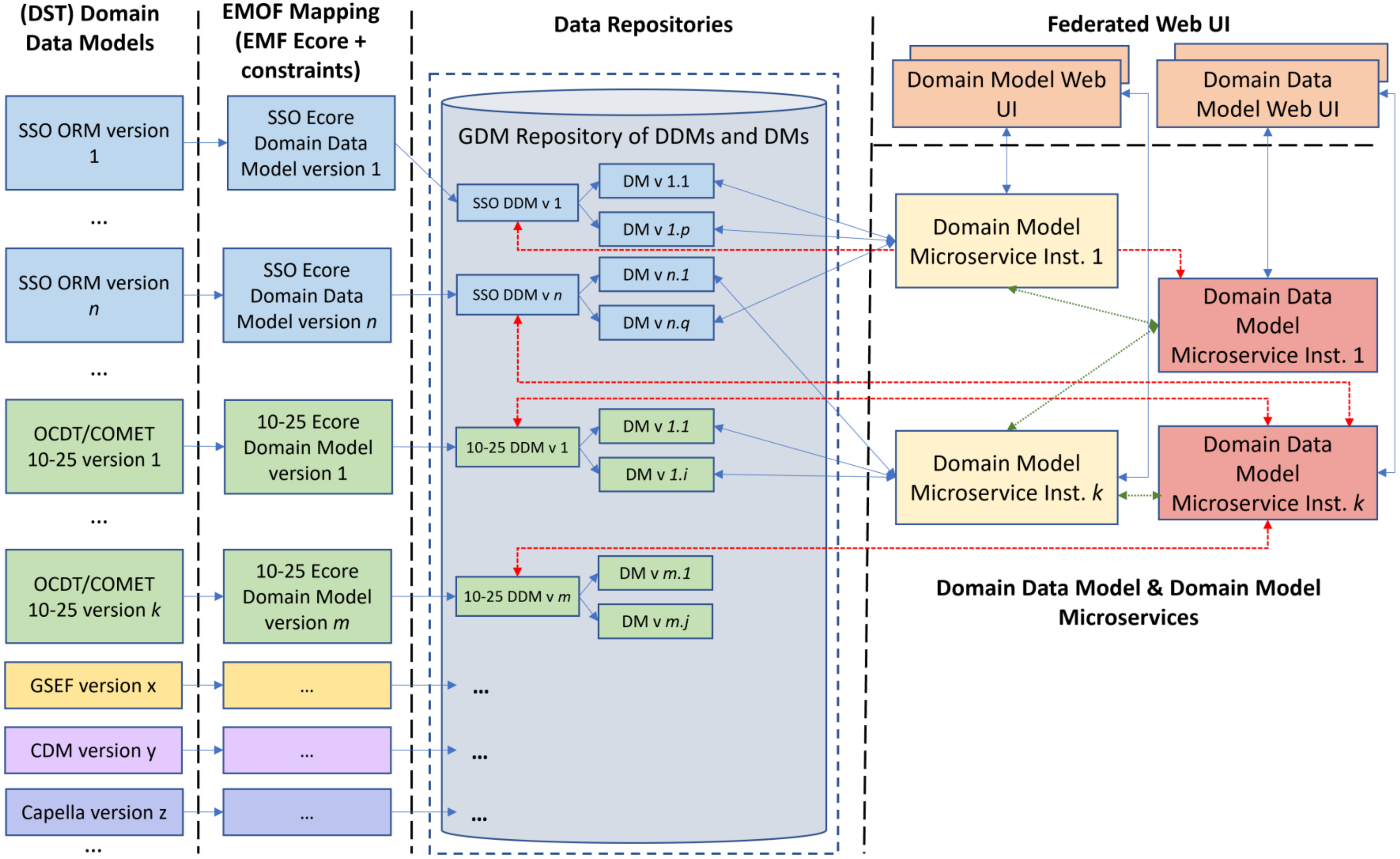
# 5. System concept

- Major evolution of Ground Segment Engineering Framework (GSEF)
  - Based on lessons-learnt
  - Using different data management approach
- Critical infrastructure aspects relevant also for MBEH



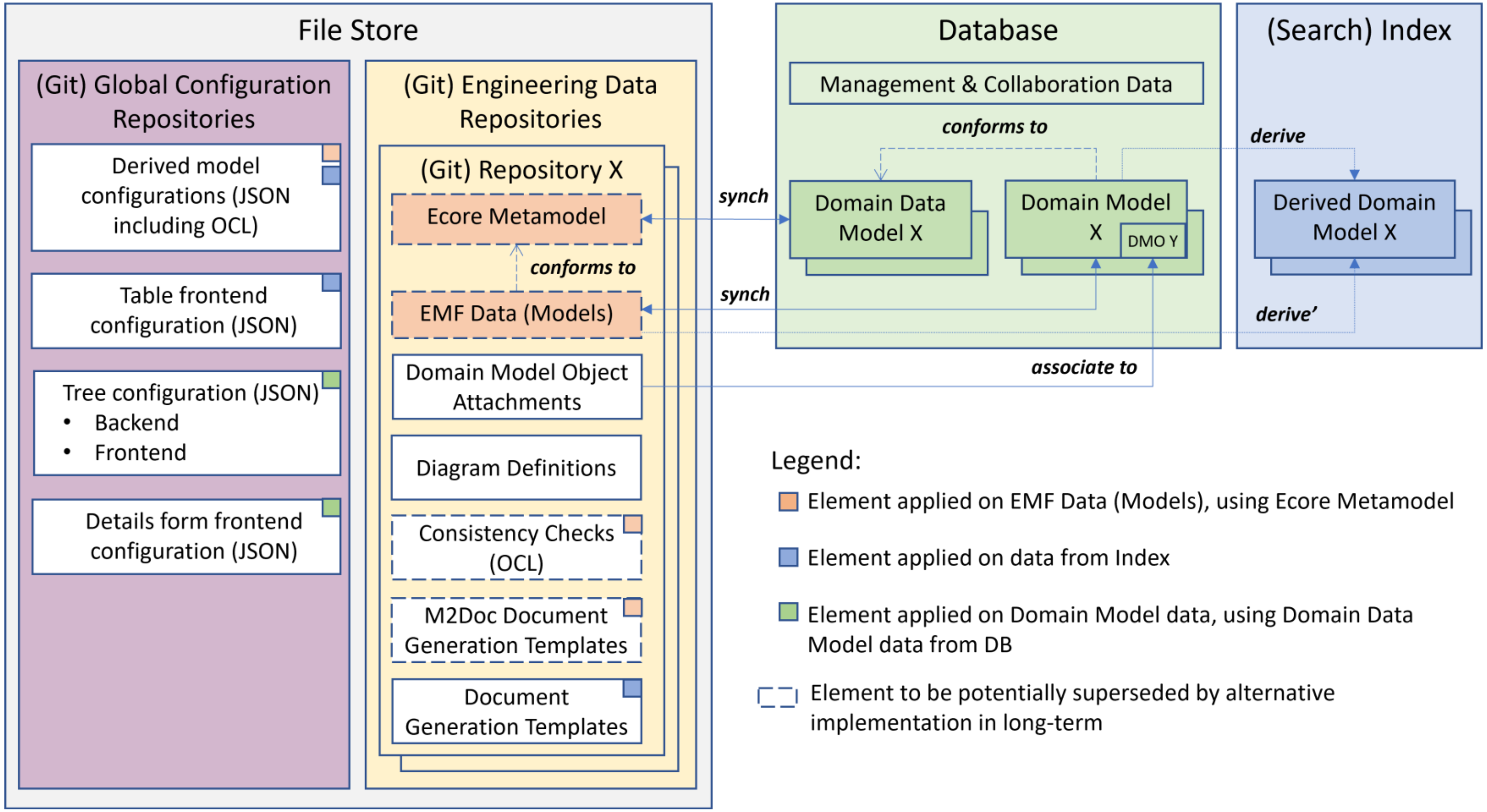
# 6. Main achievements – data model levels

- Highly abstract, Generic Data Model (GDM) used as meta-metamodel to specify other Domain Data Models (DDMs) for various engineering domains
- GDM based on the Eclipse Modeling Framework (EMF) Ecore metamodel – can be seen as reference implementation the Essential Meta Object Facility (EMOF) metamodel
- Allows using ESEP for any EMF/Ecore compatible MBSE tool or framework
- Multiple DDM versions and associated Domain Model (DM) versions supported in parallel - requirements, design, operations tailoring, verification and validation etc.



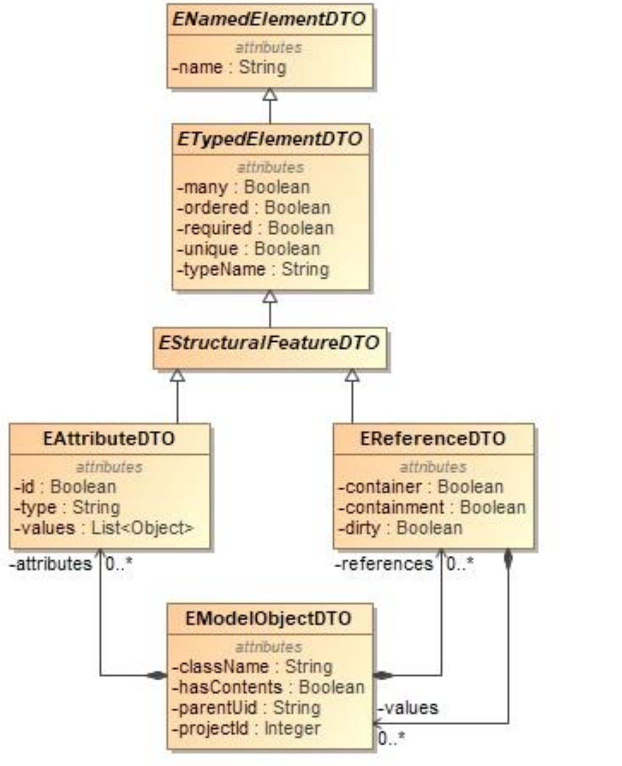
# 6. Main achievements – hybrid data management approach

- ❑ Combination of technologies to address different use cases – no “one size fits all” solution at system engineering use case level
- ❑ Database for consistency, detailed change tracking / auditing, migration etc.
- ❑ OpenSearch Index for derived models and associated views, full text search, enhanced query support, dashboards, notebooks etc.
- ❑ File stores for file management (binary files, configurations etc.)
- ❑ EMF still an option – a combination of ESEP and GSEF approach possible in same solution

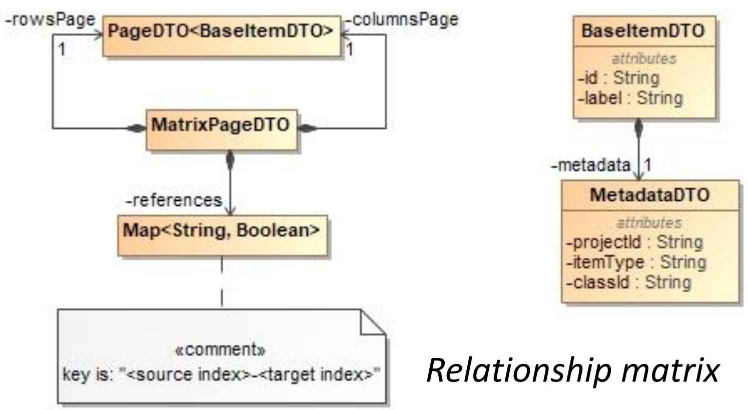


# 6. Main achievements – view-based interfaces

- View-based interfaces - retrieve only data required for constructing a specific view, e.g. tree, details form, table
  - Enhanced performance
  - Modularity and flexibility
- View data derived on backend through configurations based on DDM
- View customized in terms of display/formatting, UI controls etc. through frontend configurations
- Derived models (especially targeting bulk-data view support) constructed based on configuration and stored in Search Index
- Configurations based on JSON with associated JSON schema definitions




*Domain Model Object (DMO) details form*



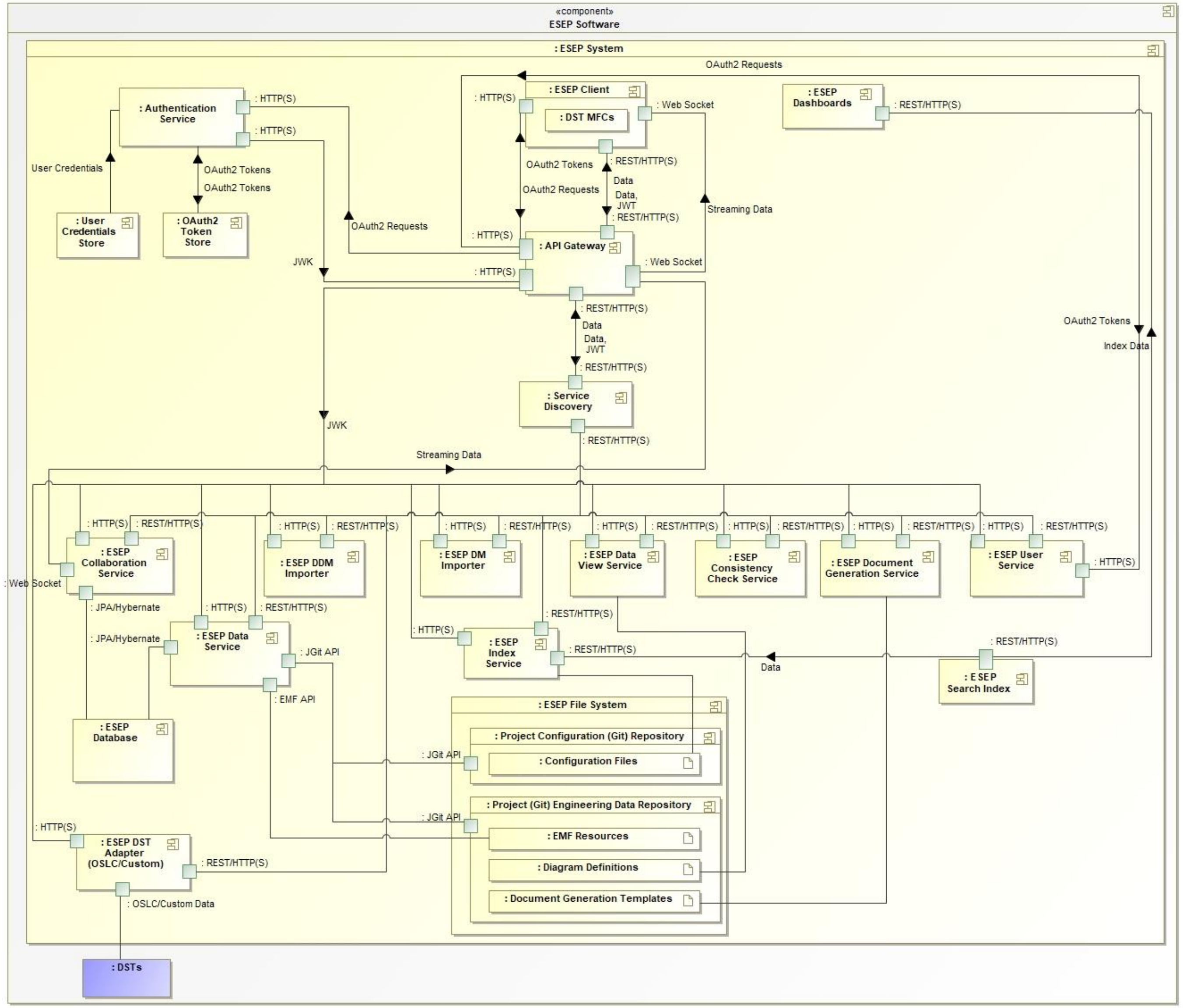

*Relationship matrix*

```
{
  "treeTypeConfigurations": [
    {
      "type": "Component",
      "labelExpression": "self.name",
      "contentReferences": [
        "children",
        "ports"
      ]
    },
    {
      "type": "Process",
      "labelExpression": "self.name",
      "contentReferences": [
        "steps"
      ]
    }
  ]
}
```

*Example DM tree backend configuration*

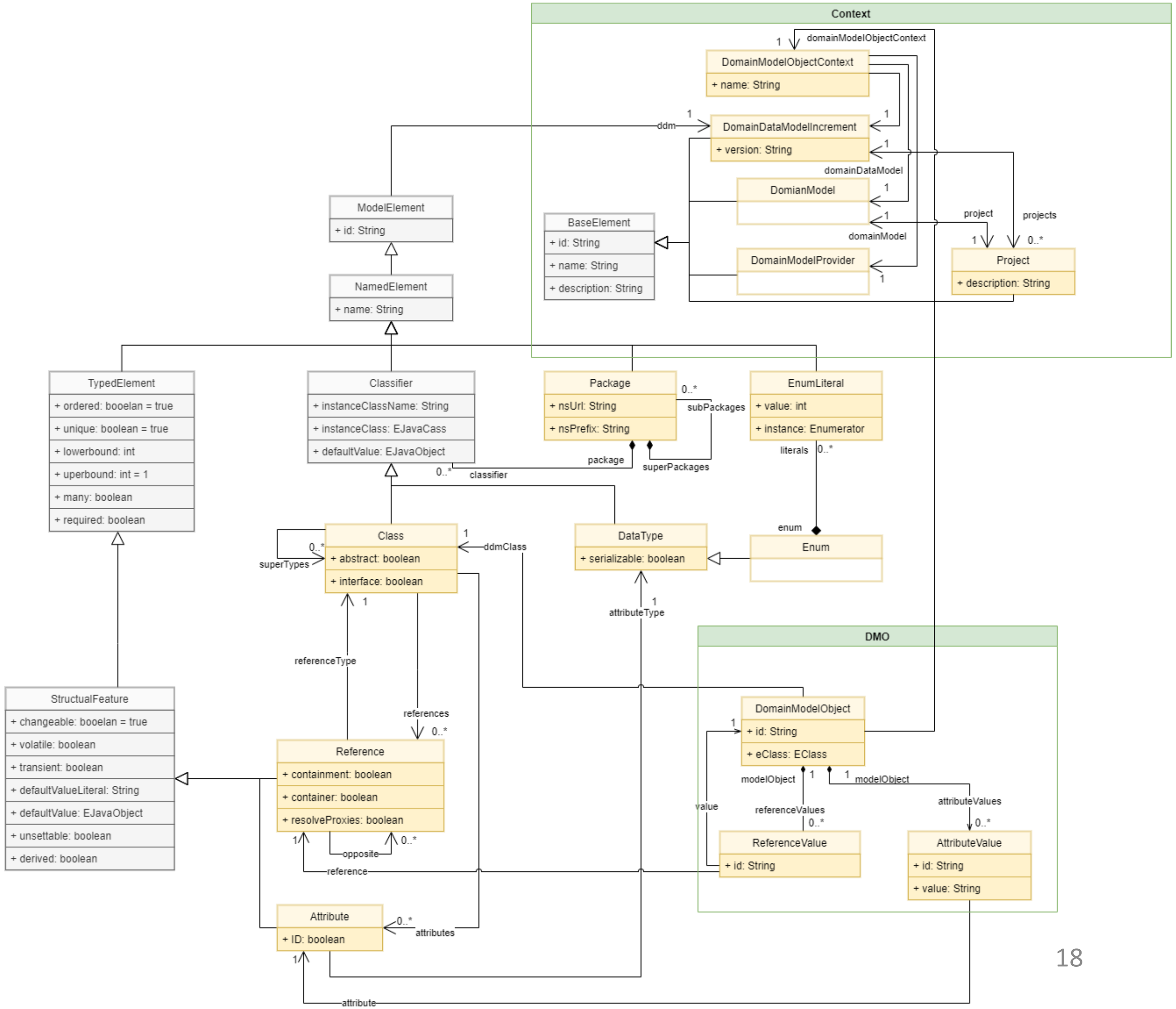
# 7. Design and implementation

- Centralized data handling through ESEP Data Service for enhanced maintainability and scalability
- Support for multiple Domain Data Models (DDMs) and associated Domain Models (DMs) in a relational database (DB) store
- OpenSearch index for derived models and associated data views support
- File store for relevant artefacts, e.g. binary or configuration files
- APIs based on HTTP/REST
- Security based on OpenID Connect (OIDC) and OAuth2



# 7. Design and implementation

- Generic Data Model (GDM) implemented in DB schema covering DDM, DM, DMO, using MariaDB as DB management system
- OpenSearch used for Search Index with associated support for dashboards, notebooks etc.
- Backend based on Spring (Boot, Cloud, Data etc.) featuring service discovery, client-side load balancing, gateway etc.
- Frontend based on Angular 12 and PrimeNG 12 as widget library
- Keycloak as authentication provider
- Detailed design as Swagger APIs generated form source code
- Docker (Compose) based deployment



# 8. Evaluation

- Test cases defined along key functional areas
- Workshops (WebEx based) for hands-on execution of test cases by representative system engineering end users
- Key features evaluated, feedback and areas for improvement captured as well as software problems to be resolved

| Relationship matrix | | | | | | |
|---|---|---|---|---|---|---|
| ESEP-TC-0300 (SE) | Relationship matrix - overview | Logged-in user with access to a project; DM data for the given project in the master database; Selected 'Relationship Matrixes' link in project-level navigation bar. | Opened relationship matrixes overview page | When the user selects the 'Relationship Matrixes' link in the project-level navigation bar, the relationship matrixes overview page is shown. When there are no matrix categories and matrixes defined, the user can create a root-level matrix category such through a dedicated 'Add' button. When there are matrix categories and possibly also matrixes, a tree with the categories and matrixes is displayed. | UC#1, UC#2, UC#6, UC#7 | Successfully covered – manual test by representative end user. |
| ESEP-TC-0310 (SE) | Relationship matrix – category creation | Logged-in user with access to a project; DM data for the given project in the master database; Opened relationship matrixes overview page; 'Add' button at top of the page selected; | Opened dialog for creating a relationship matrix category; New relationship matrix category in matrix tree in overview page upon matrix category creation. | When the user selects the 'Add' button in the relationship matrixes overview page, a dialog for creating a new relationship matrix category is displayed. After the required details for the category – name and description are populated and the 'Save' button is clicked, the category is stored in the master database and added in the matrix tree in the relationship matrixes overview page. | | Successfully covered – manual test by representative end user. |
| ESEP-TC-0320 (SE) | Relationship matrix – sub-category creation | Logged-in user with access to a project; DM data for the given project in the master database; | Context menu for adding a matrix sub-category displayed; Dialog for creating a relationship matrix sub- | When the user selects the 'Add Sub-Category' menu in the relationship matrixes tree context menu on a category tree node, a dialog creating a new relationship matrix sub-category is displayed. | | Successfully covered – manual test by representative end user. |

# 9. Conclusions

- ESEP designed and implemented with the Model Based Engineering Hub (MBEH) use cases in mind – as an engineering data integration platform

- GSEF used as conceptual starting point, but system implemented largely from scratch, based on a completely different data management approach

- ESEP embraces the concept that there is no ‘one-size-fits-all’ solution not only at a level of a Domain Data Model (DDM), but also at the level of engineering use cases in general

- ESEP offers flexible data management approach, allowing utilization and co-existence of multiple EMOF/object-oriented DDMs and versions thereof and associated Domain Models (DMs) and versions thereof - requirements, design, operations tailoring, verification and validation etc.

- Primary use case is to have a central DDM based on the Space System Ontology (SSO), but other DDMs are also supported, fostering MBSE adoption while the work on the SSO is still on-going

# 10. Next steps

- ESEP to be used as starting point for the MBEH development in MBEH consortium where SpaceCube is primary responsible for technical implementation

- Possible (GSEF equivalent) extensions to be considered, pending requirements elaboration and prioritization under MBEH:
  - Configurable export to Microsoft Excel
  - Consistency checks
  - Document generation
  - Collaboration support
    - Ad-hoc discussions
    - Labels
    - Push-notifications
    - Change review support
    - Formal (ECSS) milestone review support

SPACE CUBE

contact: services@space-cube.de