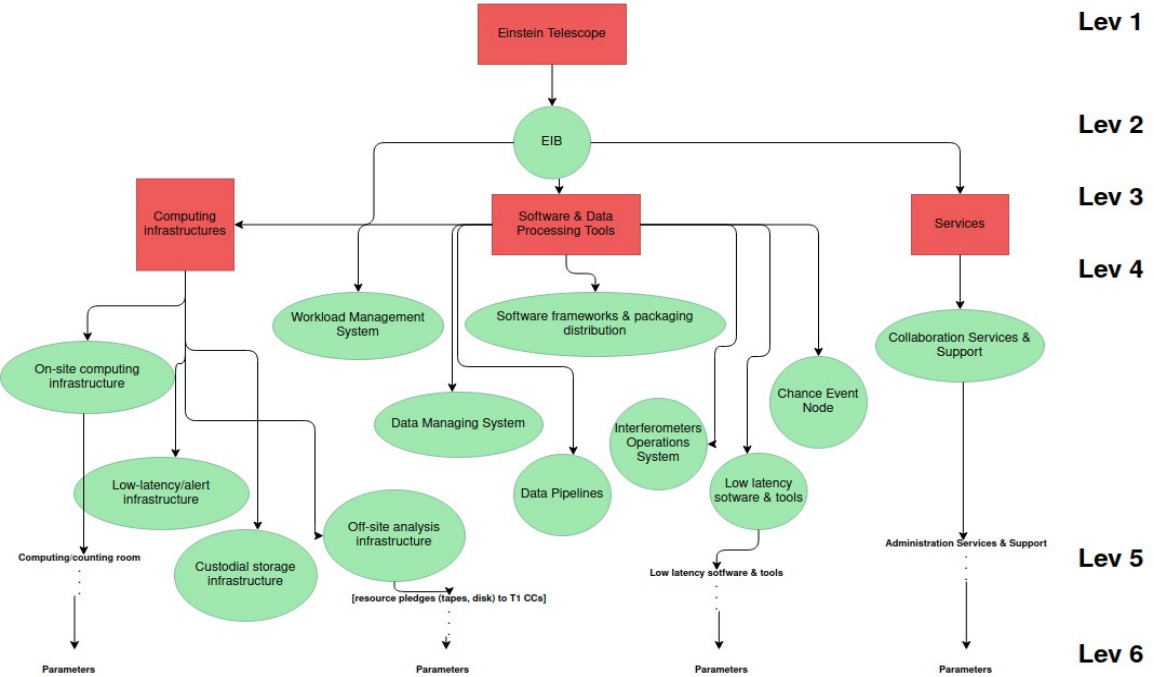
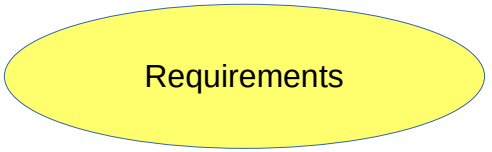
Document ID	TBA
Document type	Slides
Document Status	
Author	O. El Mecherfi
Verified by	
Approved by	

ET PBS Database Status

Introduction

- The PBS database (DB) is a critical part for managing and visualizing PBS elements and their associated parameters. The initial population of the database was carried out using Excel files provided by system coordinators for each ET system.





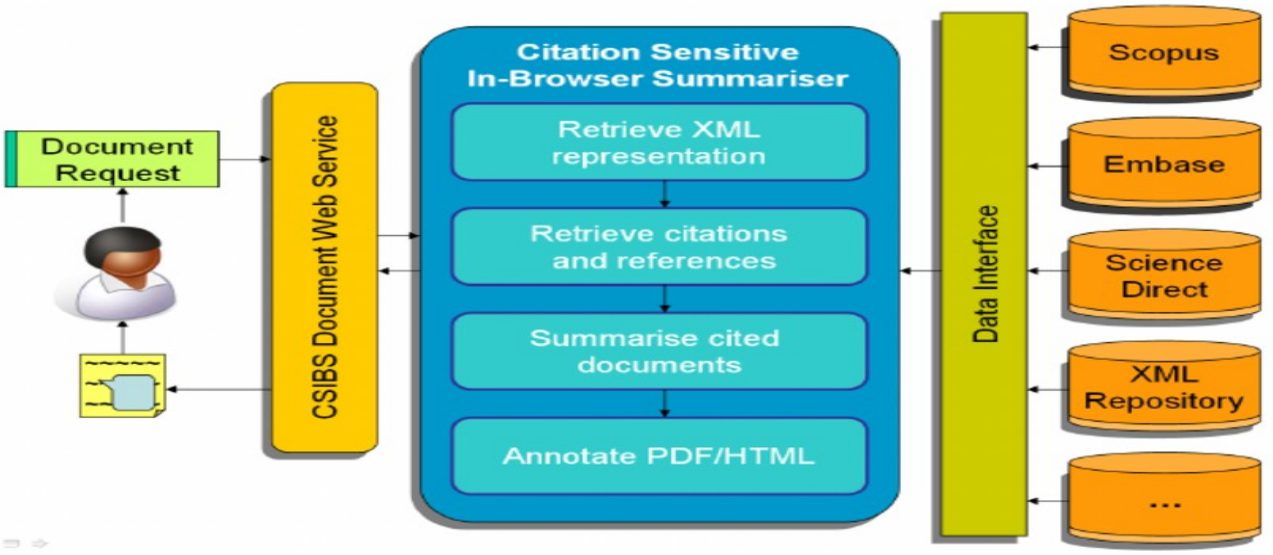
Purpose and Objectives

To allow ET collaborators to visualize PBS elements and their parameters.

To implement and maintain the hierarchical structure of the PBS within the DB.

To provide an intuitive and user-friendly interface for navigating PBS elements based on parent-child relationships.

System Architecture



Current Status of the Database

Initial Data Input : Successfully imported from Excel files.

A process from converting some tree of elements, we managed to make it to work,

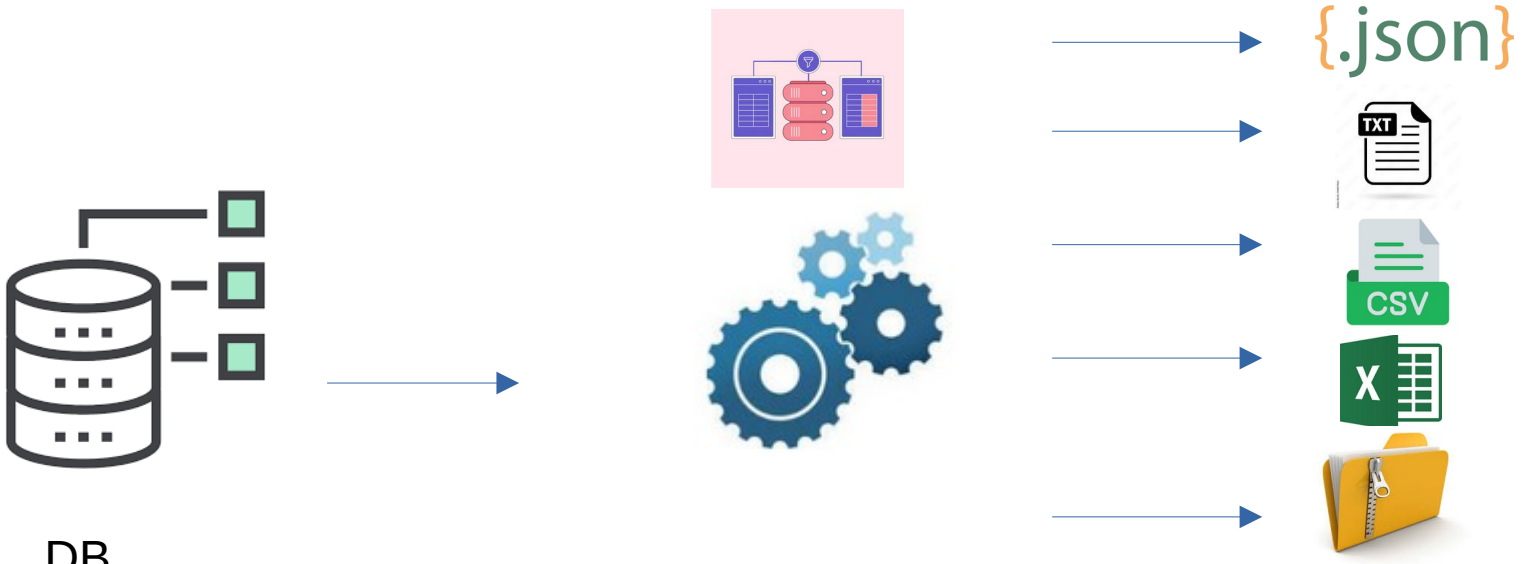
If we ever change an element parent, or add a new child, we take care of the track process history,
and also not break everything !

The admin can upload a bunch of files(400 .xlsx files) at runtime, we made sure that the stream upload work as fast as possible,
With a feedback log from any files that didn't was inserted to the DB, why ? The file structure may be not respecting our rules :

the type of possible errors are :
1) - files **PBS ID** mismatch with the **pbs code** file name
2) - files **PBS ID** missing or misspelled - file pbs_unit_name '**pbs_unit_name**' and '**pbscode**'
3) - file '**pbs_unit_name**' and '**pbscode**' doesn't have element ==> means the file didn't an associated element from the pbs tree !
4) - file functional parameters columns misspelled or an-ordred for file with pbs_unit_name '**pbs_unit_name**' and **pbscode**'
5) - file Integration parameters columns misspelled or an-ordred for file with pbs_unit_name '**pbs_unit_name**' and **pbscode**'

Allowing Exports and Filters

Show and Export the database with filtered Queries Included



DB

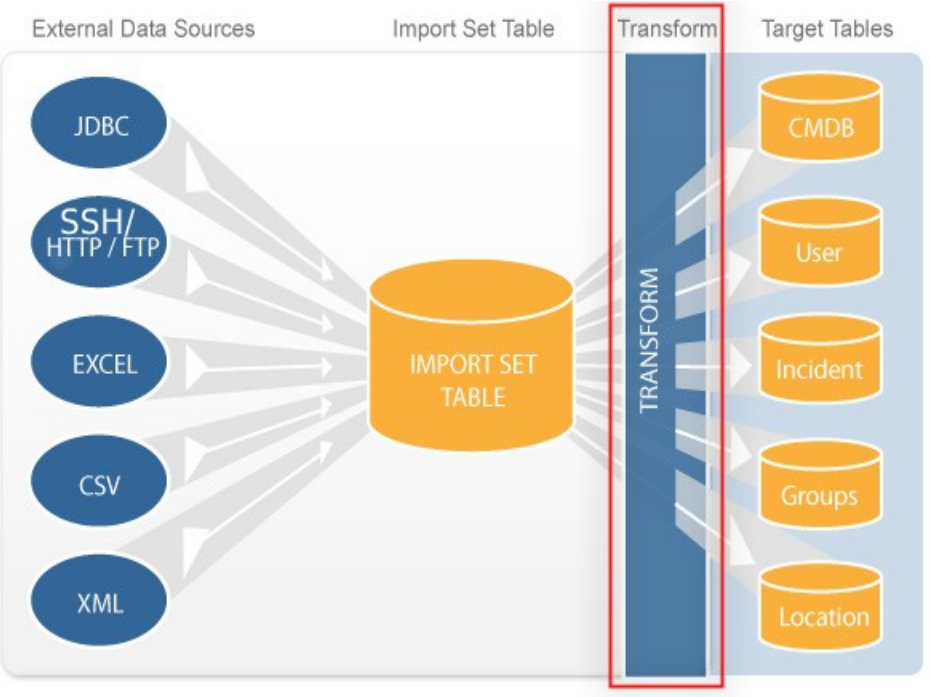
Current Status of the Database

Initial Data Input : Successfully imported from Excel files.

Structural Integrity : The PBS hierarchy is well-defined and fully operational.

User Roles :

- **All ET collaborators** : Read-only access.
- **System managers** : Delete/Write access to manage their sections.
- **Experts** : Management rights for administrative tasks.



USER ROLES

Role	PBS rights	RACI	Description
Technical coordinator	RWP	R	Responsible for the PBS editing and modification. Responsible for all process validation
Unit manager (level n)	RP	A	Responsible for the unit breakdown and for change request proposal
Unit manager (level n+1)	RP	A	Responsible for the unit breakdown and for change request proposal verification
System manager	RP	A	Responsible for the unit breakdown and for change request proposal validation
e-Infrastructure manager	RW	C	Responsible for the PBS e-Infrastructure management tool
Quality manager	RP	A	Responsible for the PBS quality process
Configuration manager	RP	A	Responsible for the PBS integration as configuration item (CI). Responsible for the change process definition
Review manager	R	I	Responsible for the PBS verification and validation reviews.
Safety manager	R	I	Responsible for PBS safety integration
Resources manager	R	I	Responsible for the PBS associated resources and budget

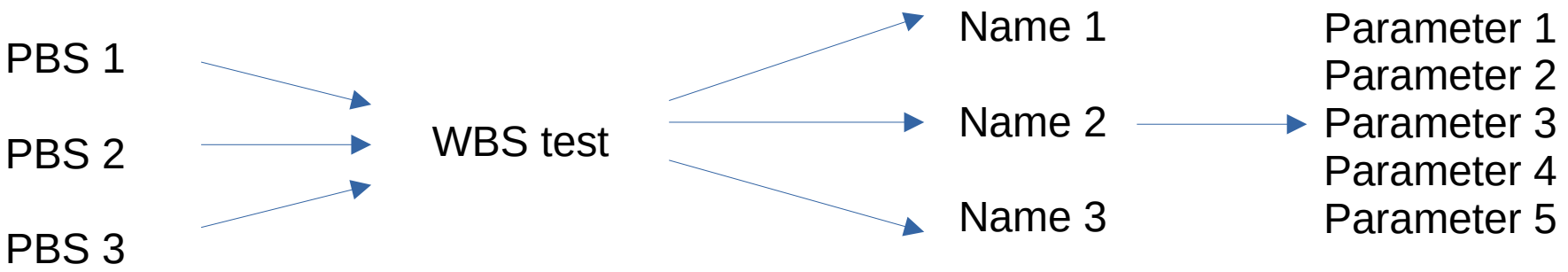
The PBS management processes are described not only by their functional flowcharts, but also from the roles and responsibilities description with respect to the full PBS lifecycle. These are summarized in Table.

PBS roles description. Rights may be of the type R= read, W=write, P=Propose (the modification). RACI acronyms are the classic Responsible,Accountable, Consulted, Informed

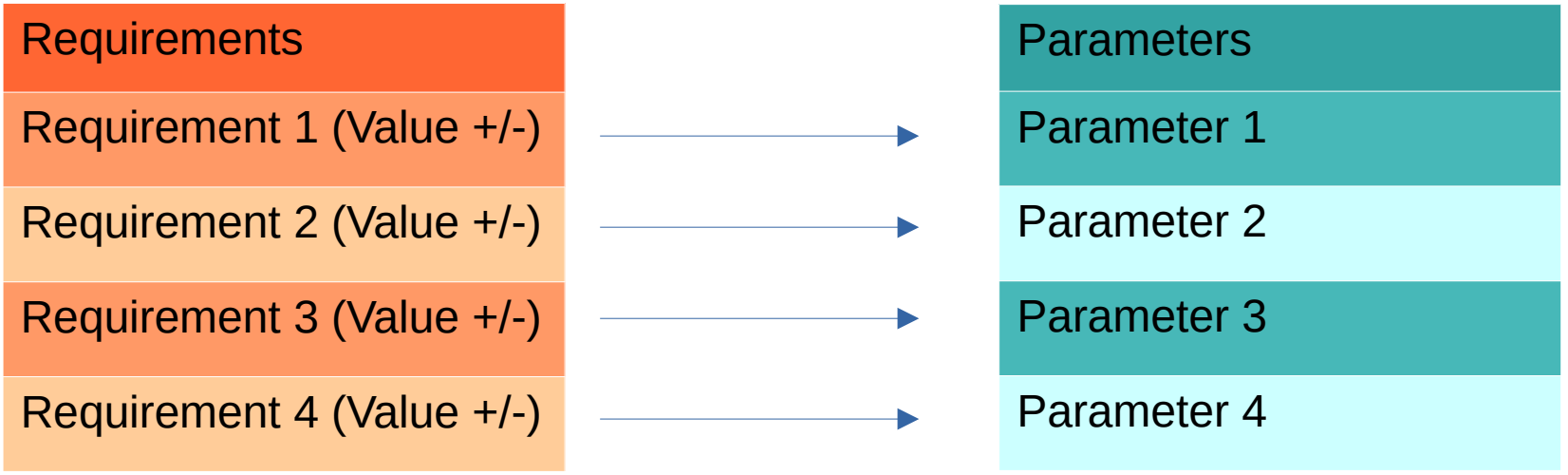
a user must have role authority to make change requests to associated elements and their parameters, change requests can only be accessed by the correct change request manager group based on a **WBS name** added as an attribute to entity elements.

WBS Groups

- A user must have role authority to make change requests to associated elements and their parameters, change requests can only be accessed by the correct group of change request managers based on a WBS name added as an attribute to the entity elements.
- This button will only appear to users belonging to a group associated with a WBS.



PBS component <an Element example>



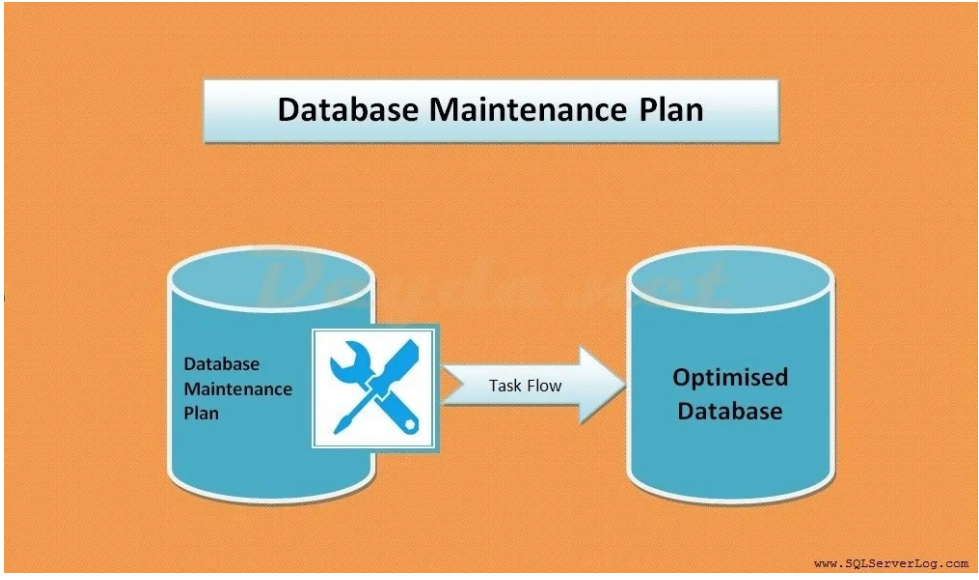
Each Parameter is constrained to some requirement value range
Requirements: These define the acceptable range for parameters or values that components must adhere to.
Parameters: Variables or characteristics of components that can be constrained by requirements.
Components: Items or modules that have parameters, which can be changed through change requests.
Change Requests: These track the changes made to parameters or components, along with metadata (like who made the change, when it happened, and what was changed).

Current Status of the Database

User Interface : Functional and user-friendly, supporting efficient data navigation.

Allowing Exports and filters : Full export in different format, and filtred requests supported.

Maintenance and Updates : Regular updates required to align with evolving project needs.

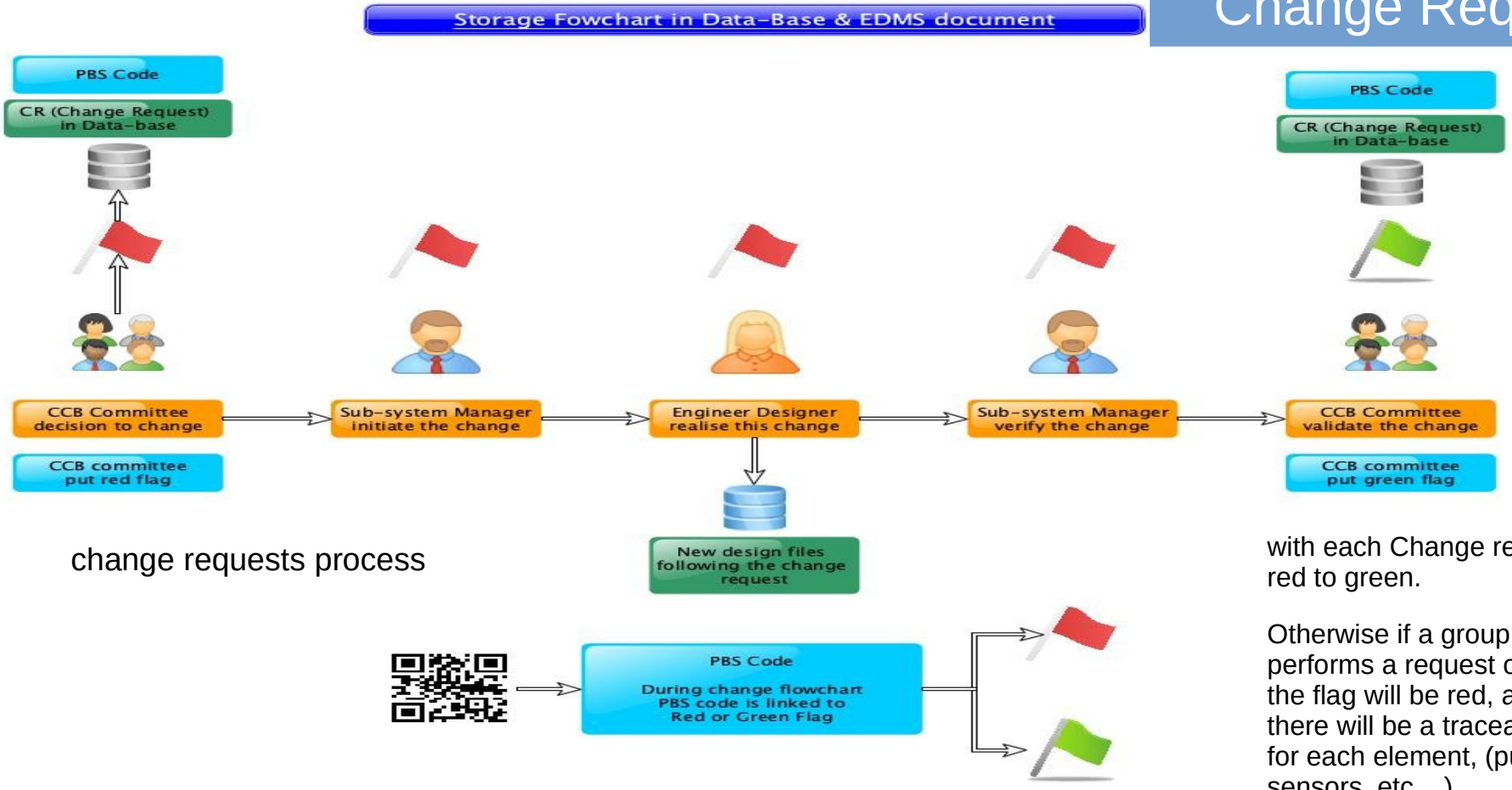


What was done ?!

- Provide first definition of parameters (functional + interface)) for all PBS items
- Connect PBS items (1.5k+) to respective parameters (10+)
- Track 10k+ quantities ==> change management process
- (Version Control system that tracks changes of the Backend / FrontEnd and DB)
- Allow evolution of PBS and parameters
- Propagate information(crucial in ensuring that modifications to attributes are properly documented, reviewed, and propagated to maintain the integrity of the PBS.)

- Risks Management Plan, still in development mode

Change Requests

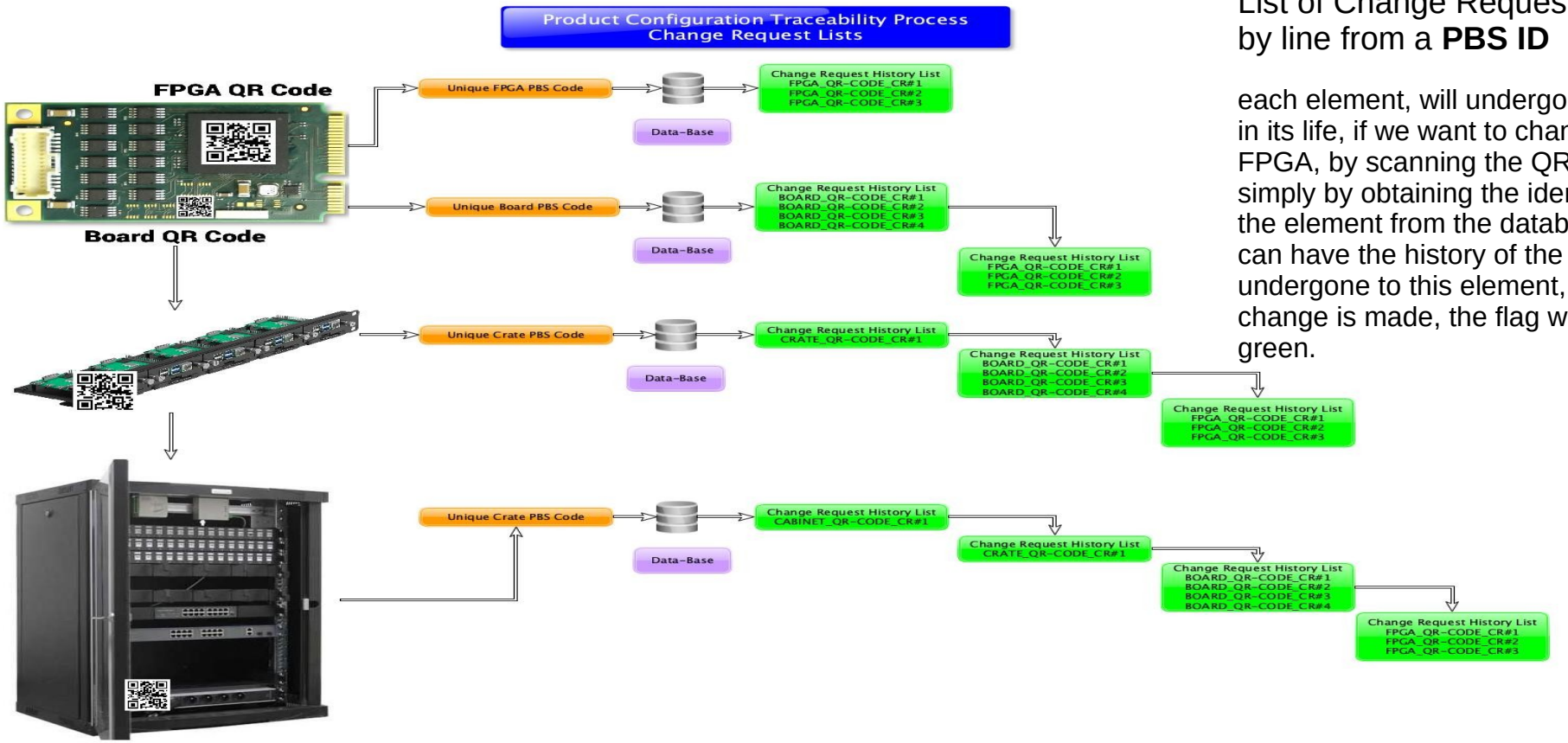


change requests process

with each Change request life, from red to green.

Otherwise if a group of users performs a request on an element, the flag will be red, and like that there will be a traceability of change for each element, (pump, newtonian sensors, etc ...)

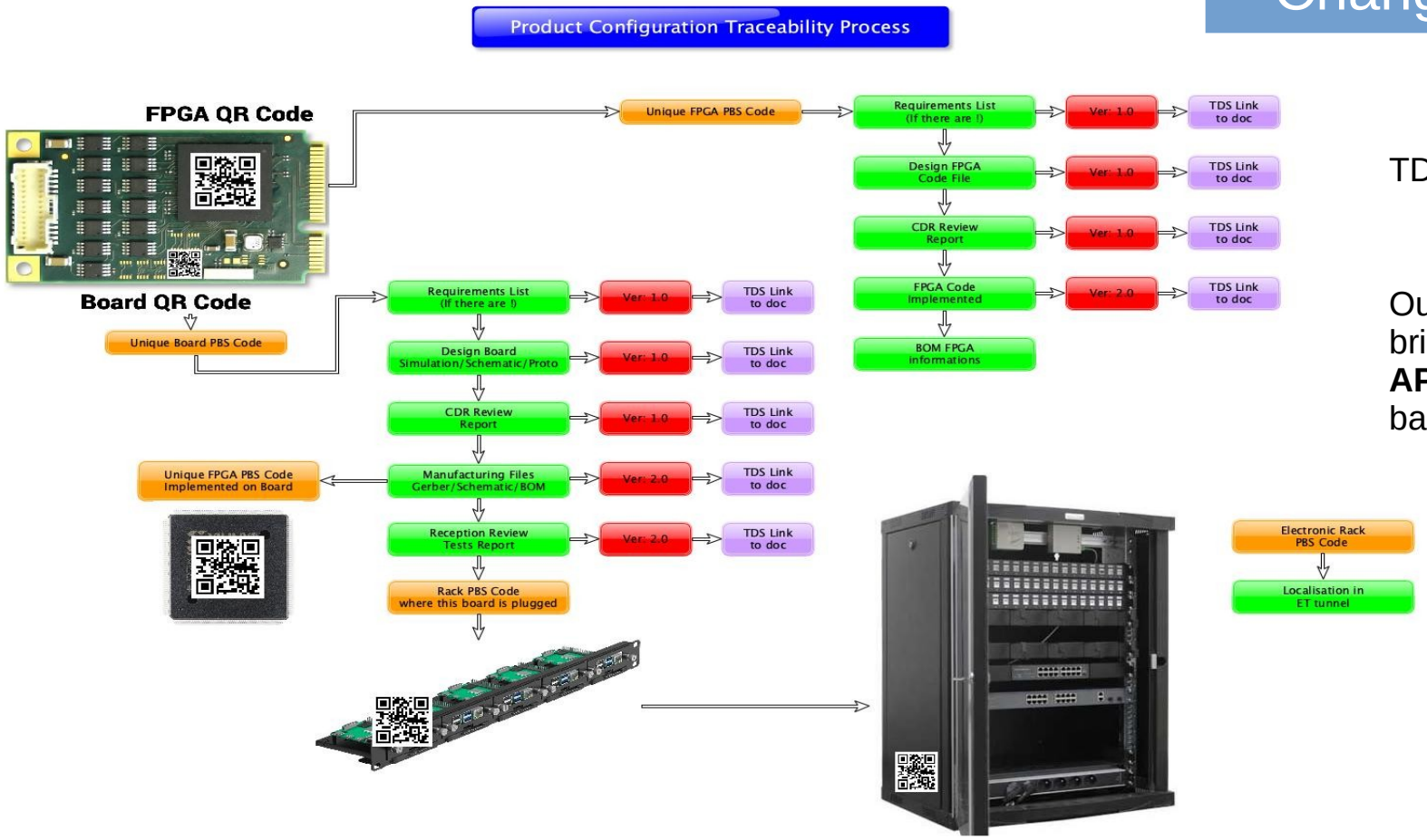
Change Requests



List of Change Requests, line by line from a **PBS ID**

each element, will undergo changes in its life, if we want to change a bar FPGA, by scanning the QR code, or simply by obtaining the identifier of the element from the database, we can have the history of the changes undergone to this element, if the change is made, the flag will be green.

Change Requests



TDS associated with Link

Our middleWare will act as a bridge to **EDMS** and **JAMA API** to get documents to view back to the user

Do We Want an Evolution in Configuration Context?

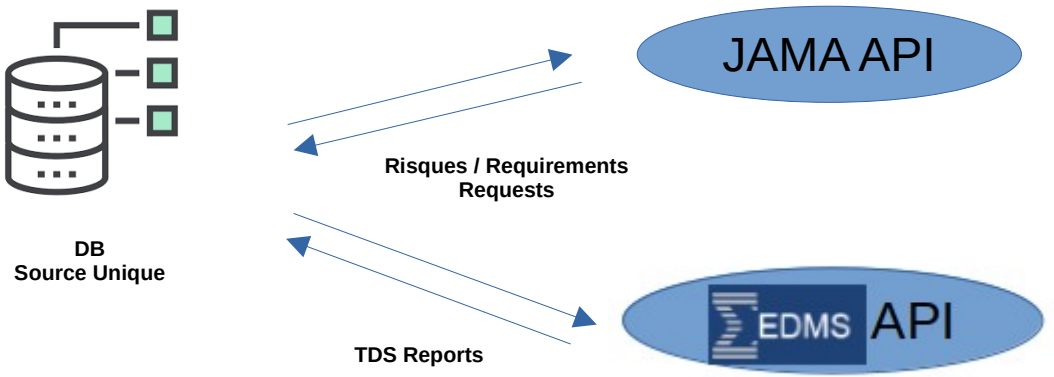
A Bridge Between EDMS, JAMA, and PBS Elements :

Our backend **(middleware)** will serve as the integration layer between:

EDMS API (Electronic Document Management System) – Storing and retrieving **technical documents (TDS, reports, etc.)**

JAMA API (Requirements & Test Management System) – Managing requirements, specifications, and test cases.

PBS Database – Storing **elements, parameters, and hierarchical relationships.**



Do We Want an Evolution in Configuration Context?

Middleware Role & Architecture

Our **middleware** will:

Fetch Documents from EDMS : Query **EDMS API** to get URLs of **TDS/reports** related to PBS elements.

Retrieve Requirements from JAMA : Pull relevant specifications and link them to **PBS elements**.

Map Data to PBS Elements : Ensure each **PBS element** has its associated documents & metadata.

Provide Secure Access : Only authorized users can access specific reports or documents.

Serve Data via API : Our **Frontend** will request documents from Django, which fetches and delivers the appropriate links or content.

Evolution Context? Registration

User Registration: Current & Future Considerations:

Current Situation:

- (for the current state) If the PBS web app currently has **fewer users**, manual registration or admin-driven account creation might be sufficient.
- However, as the ET project **grows**, a scalable user registration system is needed to handle more users efficiently.

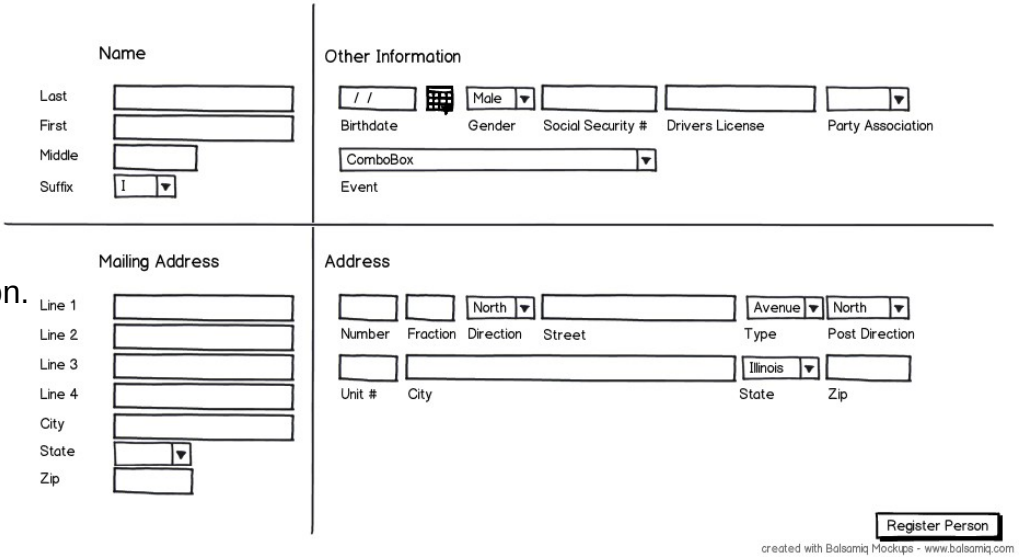
Evolution Context? Registration

Should We Develop a User Registration Form?

- Yes, developing **a self-service user registration form** for ET members would streamline onboarding and reduce administrative workload. This form should include:
- **Basic User Details:** Name, email, affiliation, role selection (ET collaborator, system manager, expert, etc.).
- **Verification Method:** Secure identity confirmation is crucial (**SMTP**).

- **Integration with Existing ET User Database:**

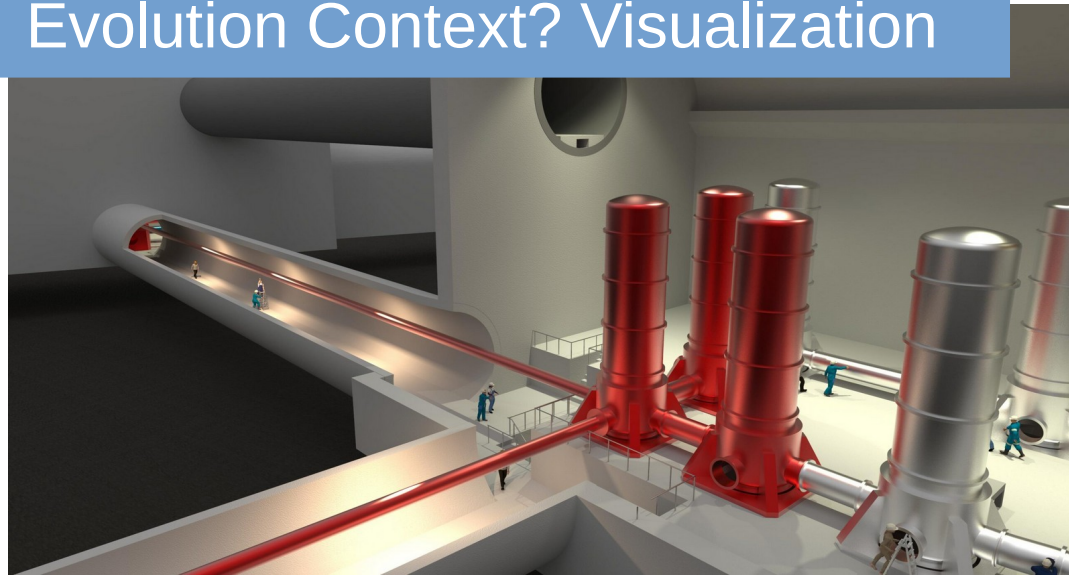
  If there's an ET-wide identity management system, integrate it instead of building a standalone user registration.

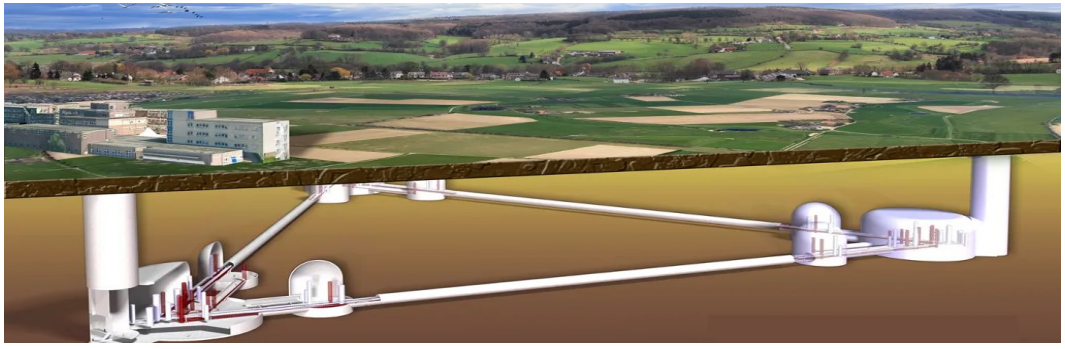


Evolution Context? Visualization

2D and 3D Visualization of Elements:

- **2D and 3D Visualization of Elements :** Enabling interactive 2D and 3Drepresentations of PBS elements to improve understanding and navigation.
- Supporting **3D model file formats file** integration and real-timerendering of elements.Interactive 2D and 3D Views: Helps users explore PBS elementsvisually rather than just through text or tables.
- **Enhanced Understanding :** A graphical representation improvesclarity, especially for complex hierarchical structures.
- **Navigation Aid :** Users can zoom, rotate, and inspect elements,improving user experience.**Real-Time Rendering:** Supports integration of 3D model formatslike **FBX, GLTF, and CAD files** for dynamic visualization.

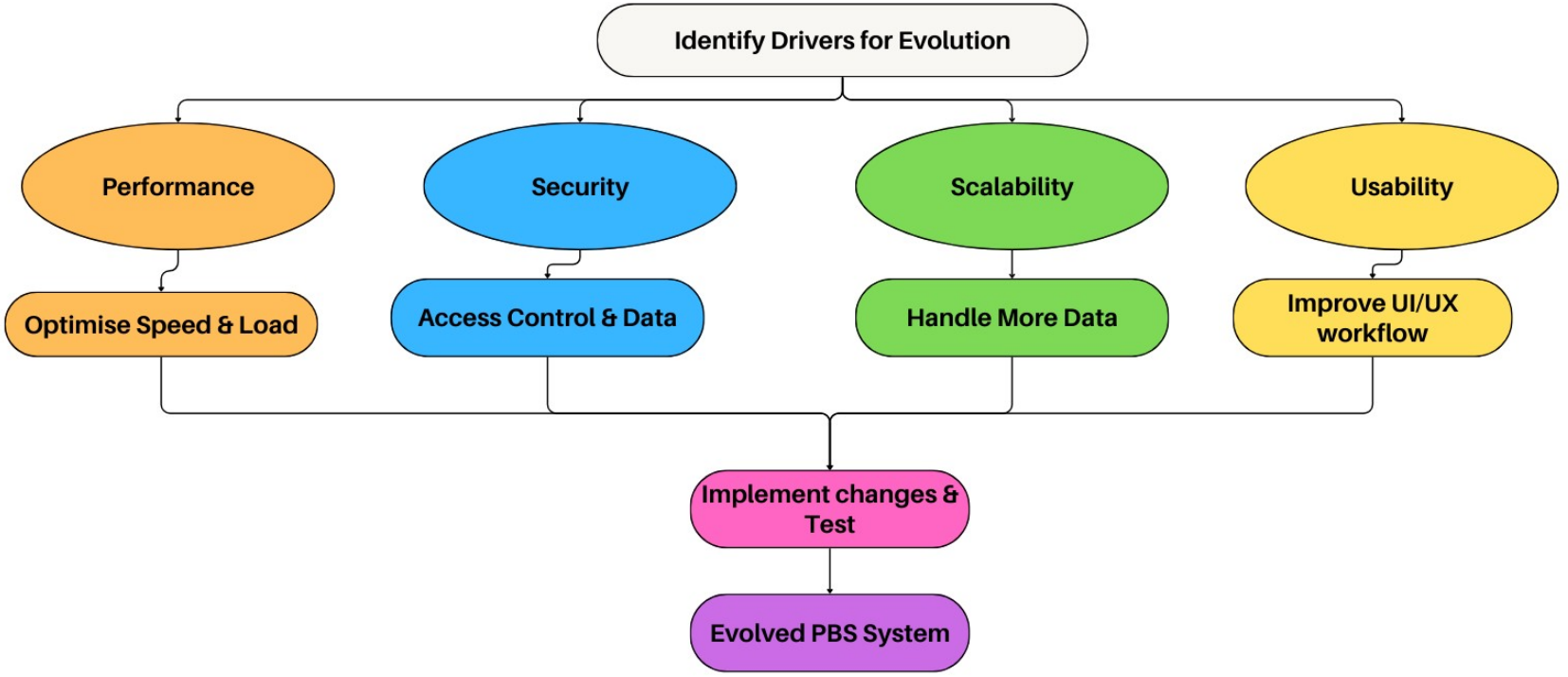




Evolution Context

Define the Evolution Context

Identify the **drivers for evolution**: Are we improving performance, security, scalability, or usability?



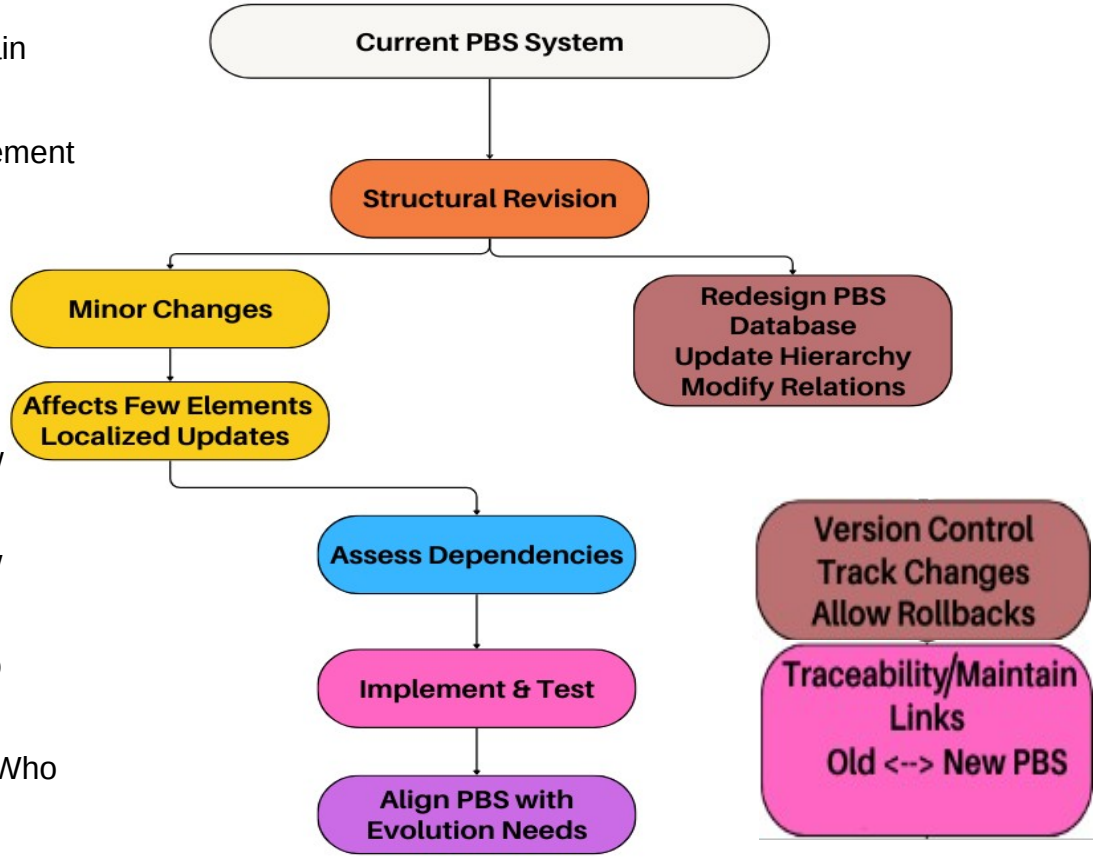
Evolution Context

Assess the **impact on the current PBS**: Will changes affect only certain elements, or is a structural revision needed?

Ensure compliance with **existing processes**, such as change management and approval workflows.

Align PBS with Evolution Needs

- **Refine the Hierarchical Structure** : If the evolution introduces new components, update the PBS accordingly.
- **Version Control** : Implement versioning to track changes and allow rollbacks when needed.
- **Traceability** : Maintain links between old and new PBS elements to understand dependencies.
- **Define roles and responsibilities:** Who validates **PBS** changes? Who implements them?



Future Outlook

Short-term Goals :

- Validate current system functionality and security.
- Optimize and Validate database queries for better performance.
- Conduct user training sessions & feedbacks, our **PBS Database Documentation** are open for support and feedbacks for modification.

Long-term Goals :

- Implement advanced analytics and reporting features.
- Introduce **AI-driven** approach / insights for predictive analysis.
- Transition to a fully cloud-based or hybrid architecture for scalability.

DB

MERCI

DB