



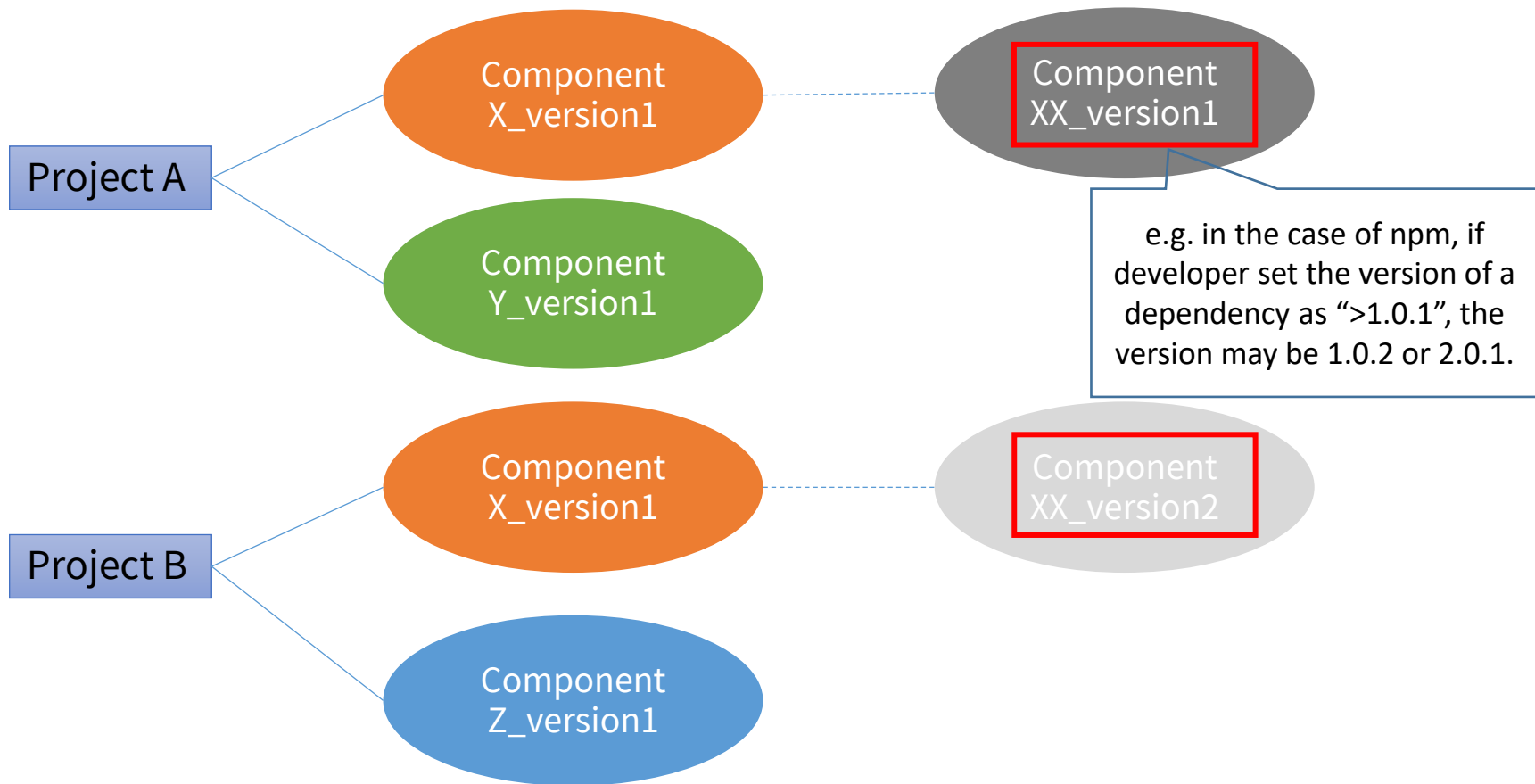
**TOSHIBA**

# A New Dependency Management Function for SW360

Toshiba Corporation  
Shi QIU  
2022/05/19

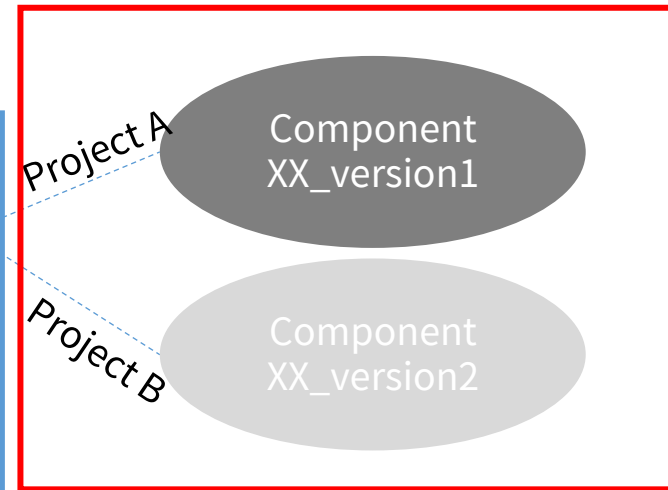
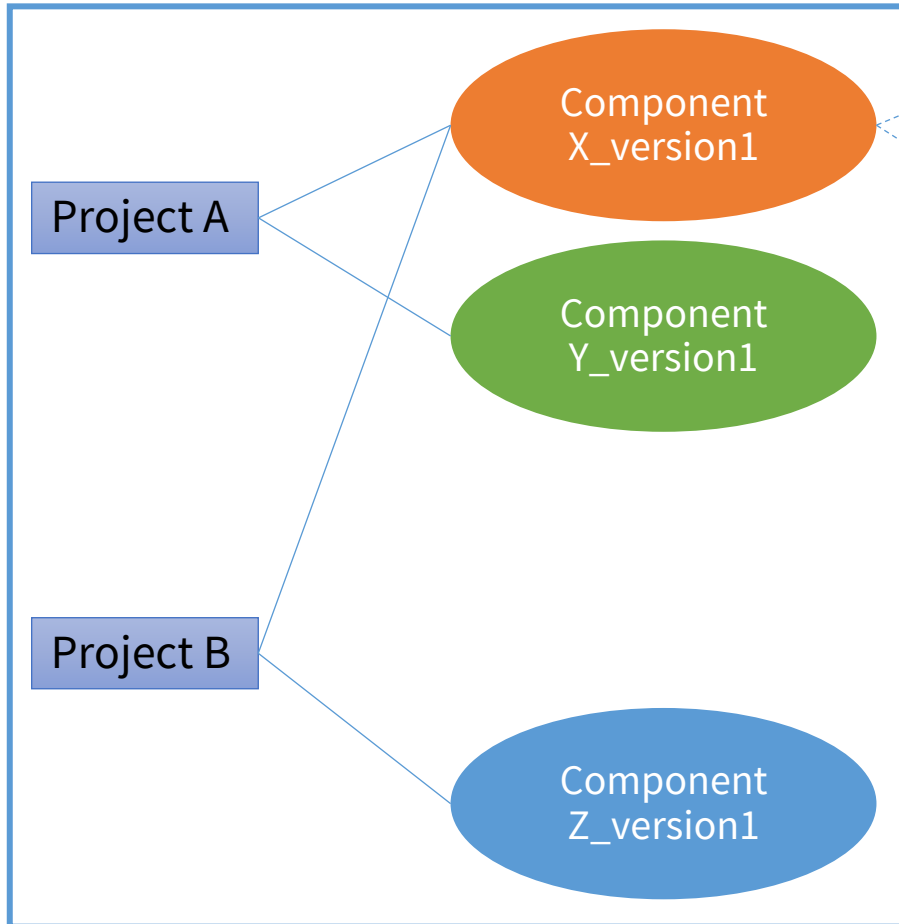
# Problem

- In practice, we face a problem that for a component with the same version, the version of its dependency may be different.



# Problem

The current situation of SW360:



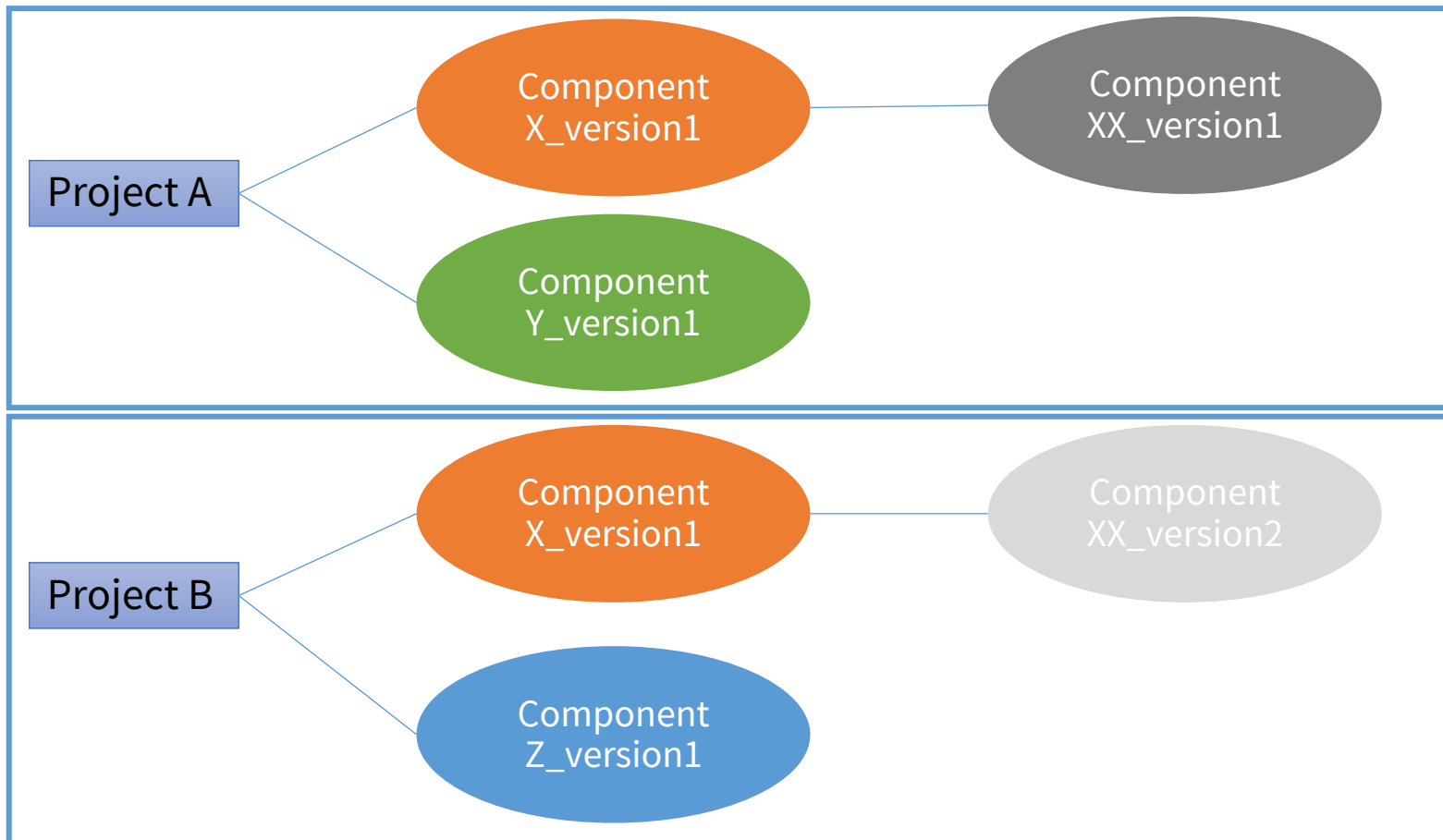
This case can not be registered in the current SW360

Only the information of the direct dependencies of a project can be registered in SW360

# Proposal

- A New Dependency Management Function for SW360:

Function allowing Project to set up its own dependency network



Both the information of the direct dependencies and transitive dependencies of a project can be registered in SW360

# Proposal

- The current situation of the “Linked Releases And Projects” page:

Summary

Administration

Linked Releases And Projects

Attachments

Obligations 0 / 0

Update Project Delete Project Cancel

PROJECT A

LINKED PROJECTS

Project name	Project Version	Project Relation ⓘ	Enable SVM
--------------	-----------------	--------------------	------------

Add Projects

LINKED RELEASES

Release name	Release version	Release relation ⓘ	Project Mainline State ⓘ	Comments
Component X	1.0	Dynamically linke ⇅	Open ⇅	Enter Comment
Component Y	1.0	Dynamically linke ⇅	Open ⇅	Enter Comment

Add Releases

“Project” can only set the direct dependencies, but can not set the transitive dependencies.

# Proposal

- The proposal of the “Linked Releases And Projects” page:

Summary

Administration

Linked Releases And Projects

Attachments

Obligations 0 / 0

Update Project Delete Project Cancel

PROJECT A

LINKED PROJECTS

Project name	Project Version	Project Relation ⓘ	Enable SVM
--------------	-----------------	--------------------	------------

Add Projects

LINKED RELEASES

Release name	Release version	Release relation ⓘ	Project Mainline State ⓘ	Comments
Component X	1.0	Dynamically linked	Open	Enter Comment
Component XX	1.0	Contained	Open	Enter Comment
Component Y	1.0	Dynamically linked	Open	Enter Comment

Add Releases

“Project” can set both of the direct dependencies and the transitive dependencies.

# Proposal

- The proposal of the “Linked Releases And Projects” page:

The screenshot shows the 'Linked Releases And Projects' page for 'PROJECT A'. It includes a sidebar with navigation options: Summary, Administration, Linked Releases And Projects (highlighted), Attachments, and Obligations (0/0). The main content area has three buttons: 'Update Project' (orange), 'Delete Project' (red), and 'Cancel' (grey). Below these are two tables: 'LINKED PROJECTS' and 'LINKED RELEASES'. The 'LINKED PROJECTS' table has columns for Project name, Project Version, Project Relation, and Enable SVM. The 'LINKED RELEASES' table has columns for Release name, Release version, Release relation, Project Mainline State, and Comments. Three callout boxes provide instructions: 1. 'Show all versions of Component. User can choose one.' points to the 'Release version' dropdowns. 2. 'Click the reload button to reload this screen and show the dependency information of the selected version of Component.' points to the reload icons in the 'Release relation' column. 3. 'The transitive dependencies can also be shown and set' points to the 'Release relation' dropdowns.

Callout 1: Show all versions of Component. User can choose one.

Callout 2: Click the reload button to reload this screen and show the dependency information of the selected version of Component.

Callout 3: The transitive dependencies can also be shown and set

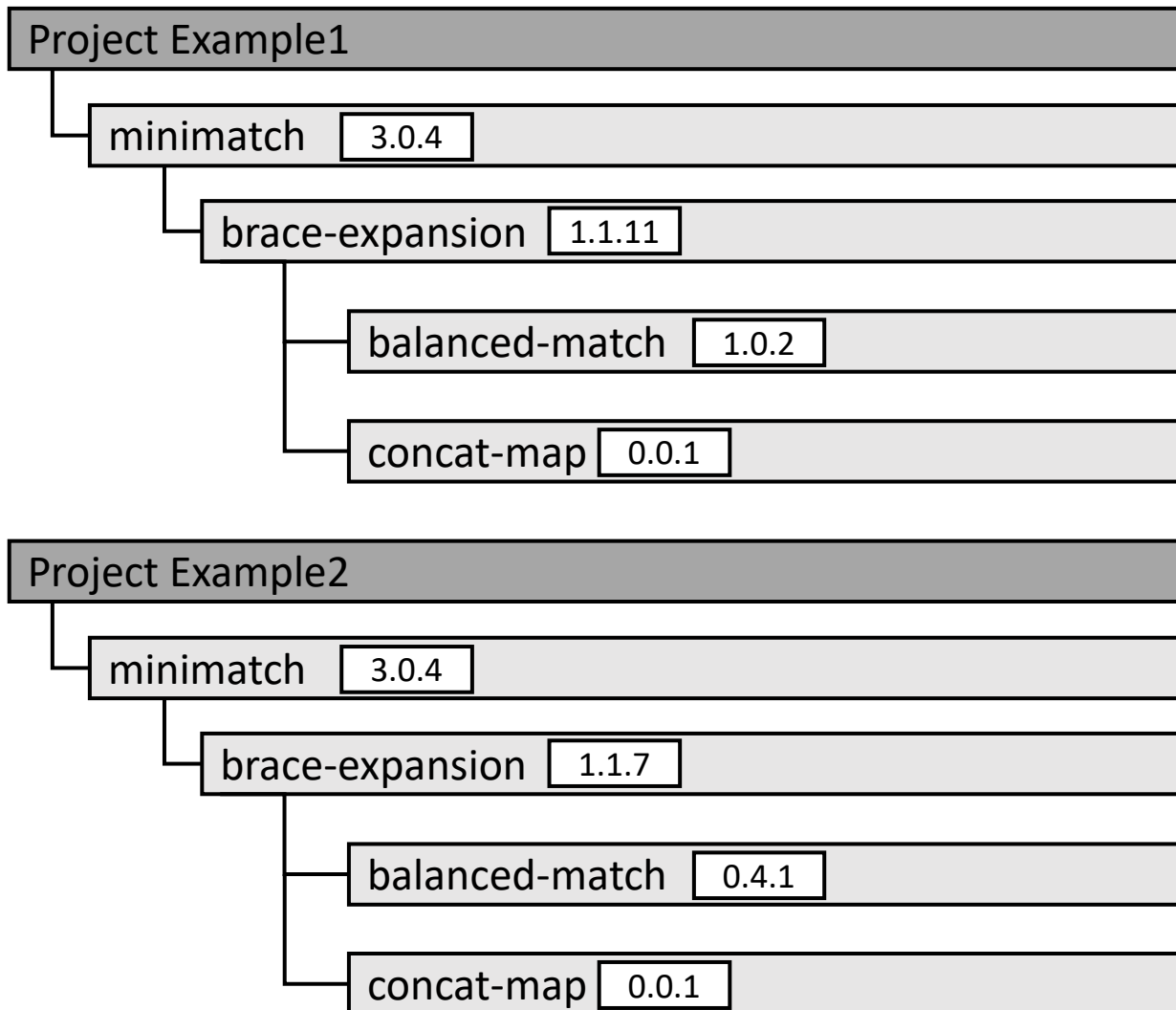
# Proposal

- Key Point:
  - Project has its own database for dependency information
    - Some dependency information (name, version, relationships between releases) will also be stored
  - The other features will be affected:
    - License Clearing tag
    - Clearing Request
    - ECC tag
    - Rest API function
    - Import/Export function
    - Change Log function
    - SPDX import



# Example

- Two projects using npm package *minimatch* as its dependencies:



In the meta file of minimatch v3.0.4, the dependency information is:

```
*****  
"dependencies": {  
  "brace-expansion":  
    "^1.1.7"  
},  
*****
```

But the dependency network of a new project (Example1) will be different with an old one (Example2).

# Example

- The current situation:

Summary Administration **Linked Releases And Projects** Attachments Obligations **0 / 0**

PROJECT EXAMPLE1

Update Project Delete Project Cancel

LINKED PROJECTS

Project name	Project Version	Project Relation ⓘ	Enable SVM
--------------	-----------------	--------------------	------------

Add Projects

LINKED RELEASES

Release name	Release version	Release relation ⓘ	Project Mainline State ⓘ	Comments
minimatch	3.0.4	Dynamically linked ▾	Open ▾	Enter Comment

Add Releases

Summary Administration **Linked Releases And Projects** Attachments Obligations **0 / 0**

PROJECT EXAMPLE2

Update Project Delete Project Cancel

LINKED PROJECTS

Project name	Project Version	Project Relation ⓘ	Enable SVM
--------------	-----------------	--------------------	------------

Add Projects

LINKED RELEASES

Release name	Release version	Release relation ⓘ	Project Mainline State ⓘ	Comments
minimatch	3.0.4	Dynamically linked ▾	Open ▾	Enter Comment

Add Releases

# Example

- The proposal:

Summary Administration **Linked Releases And Projects** Attachments Obligations 0/0

Update Project Delete Project Cancel PROJECT EXAMPLE1

LINKED PROJECTS

Project name	Project Version	Project Relation	Enable SVM
--------------	-----------------	------------------	------------

Add Projects

LINKED RELEASES

Release name	Release version	Release relation	Project Mainline State	Comments
minimatch	3.0.4	Dynamically linked	Open	Enter Comment
brace-expansion	1.1.11	Dynamically linked	Open	Enter Comment
balanced-match	1.0.2	Dynamically linked	Open	Enter Comment
concat-map	0.0.1	Dynamically linked	Open	Enter Comment

Add Releases

Summary Administration **Linked Releases And Projects** Attachments Obligations 0/0

Update Project Delete Project Cancel PROJECT EXAMPLE2

LINKED PROJECTS

Project name	Project Version	Project Relation	Enable SVM
--------------	-----------------	------------------	------------

Add Projects

LINKED RELEASES

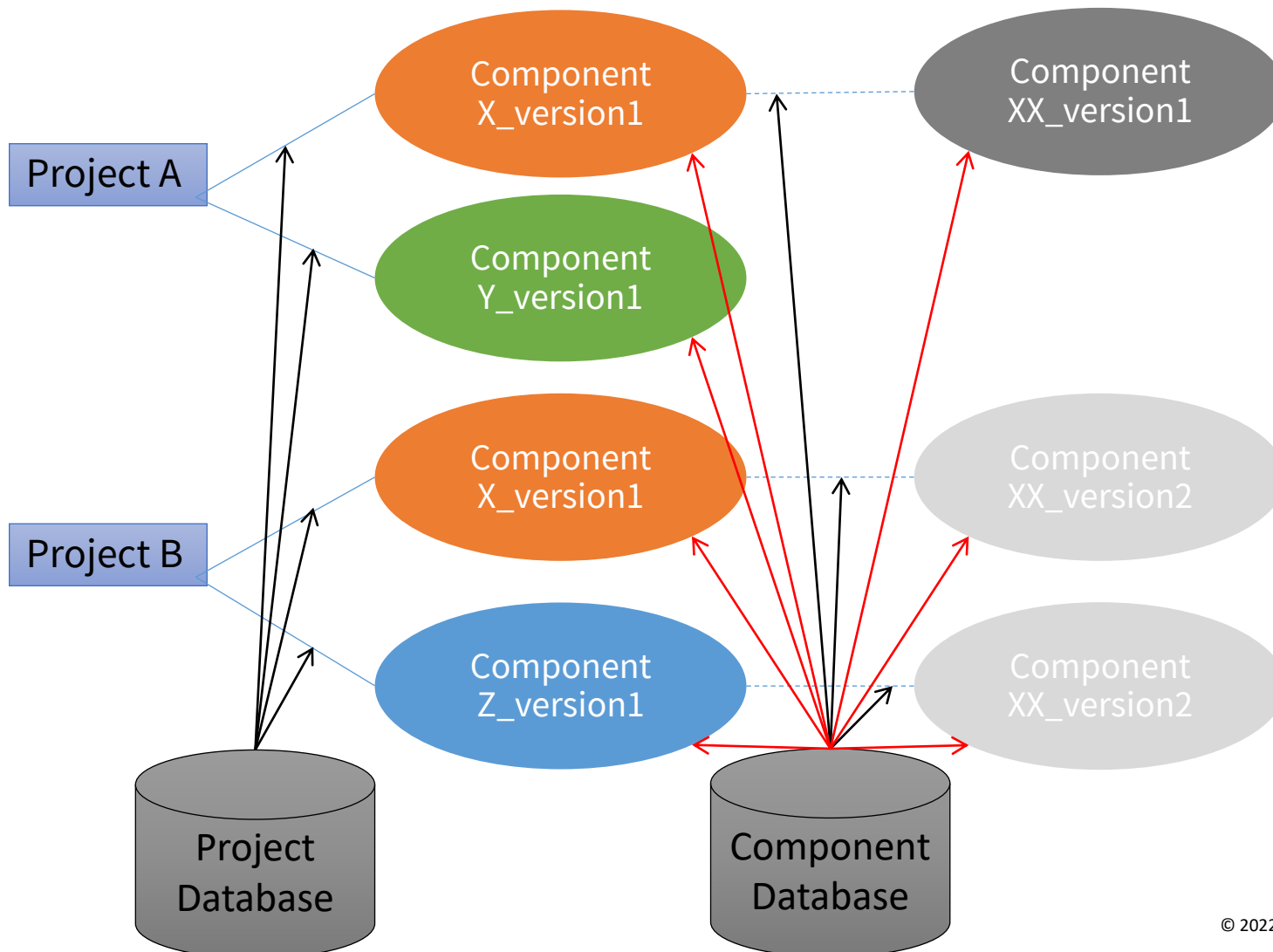
Release name	Release version	Release relation	Project Mainline State	Comments
minimatch	3.0.4	Dynamically linked	Open	Enter Comment
brace-expansion	1.1.7	Contained	Open	Enter Comment
concat-map	0.0.1	Dynamically linked	Open	Enter Comment
balanced-match	0.4.1	Dynamically linked	Open	Enter Comment

Add Releases

# Proposal

Black arrow:  
Dependency information  
Red arrow: Component  
field information

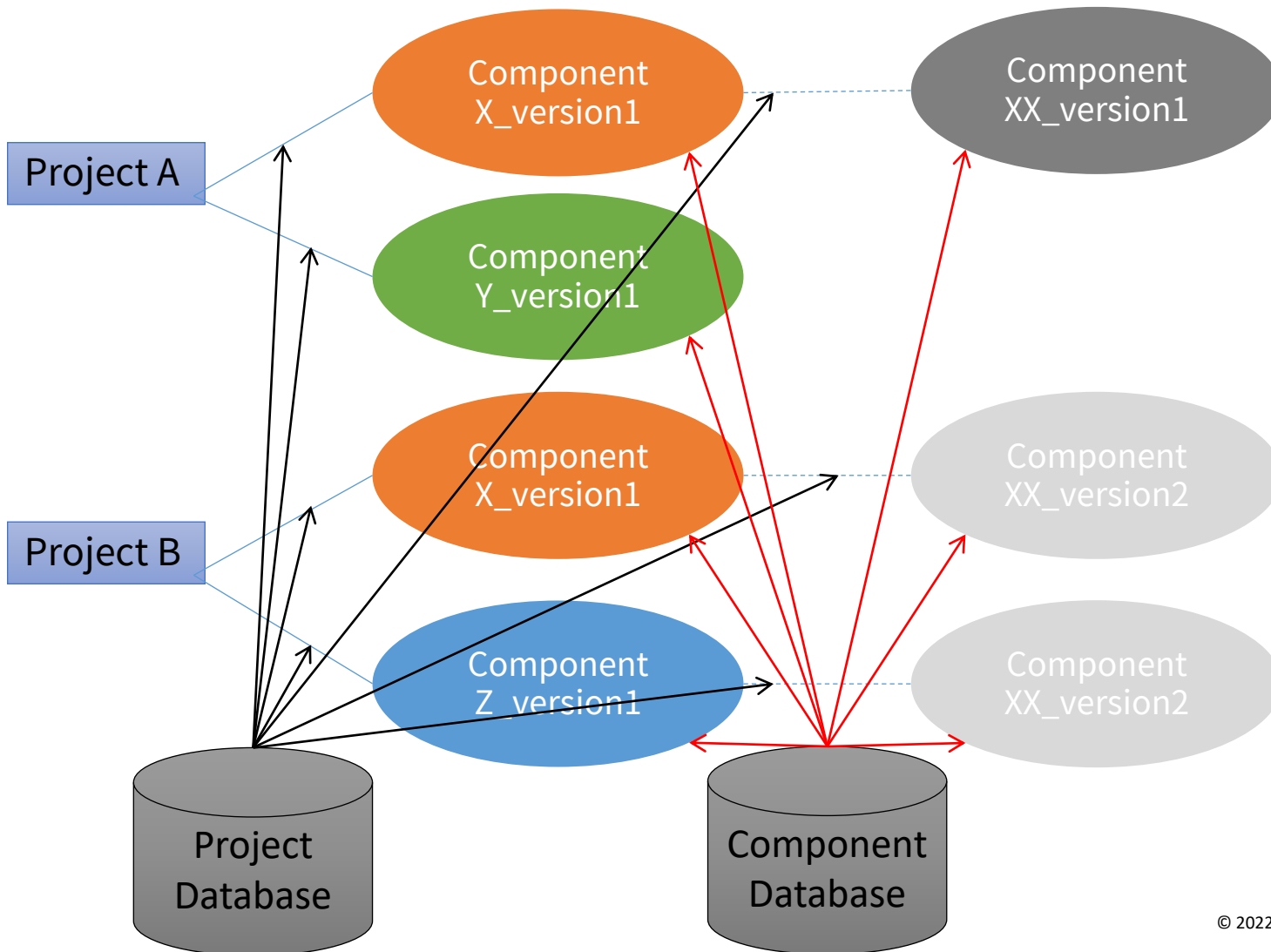
- Way of storing dependencies (current)



# Proposal

Black arrow:  
Dependency information  
Red arrow: Component  
field information

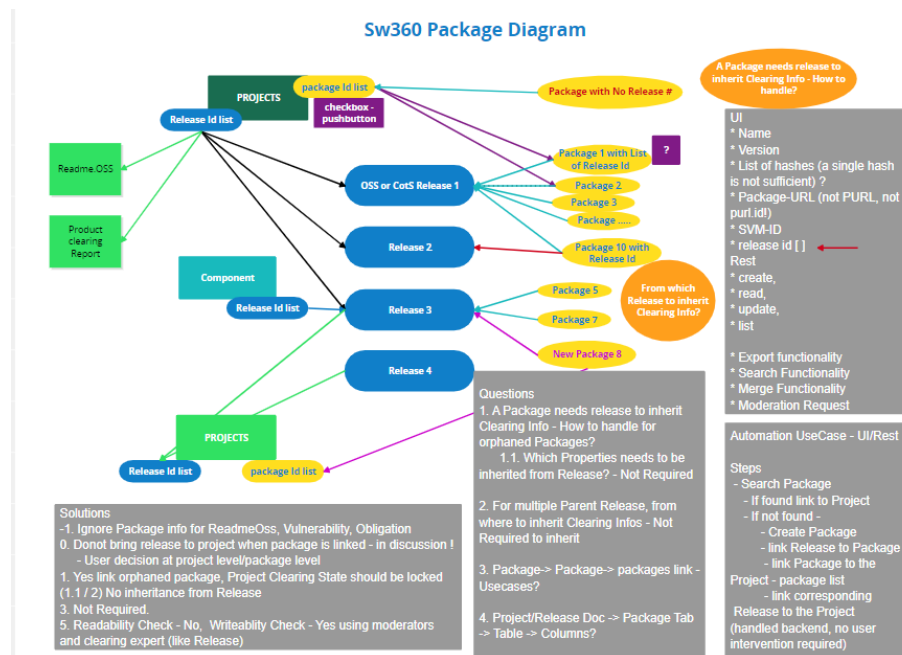
- Way of storing dependencies (proposal)



# GUI Design

- Keep consistent with:

<https://github.com/eclipse/sw360/issues/1374>



- Cooperation with ORT will also be considered after the development of this function

## Related Link

- The development branch of Toshiba:
  - <https://github.com/toshiba/sw360/tree/dev/project-and-release-dependency-management-function>

**TOSHIBA**





**TOSHIBA**

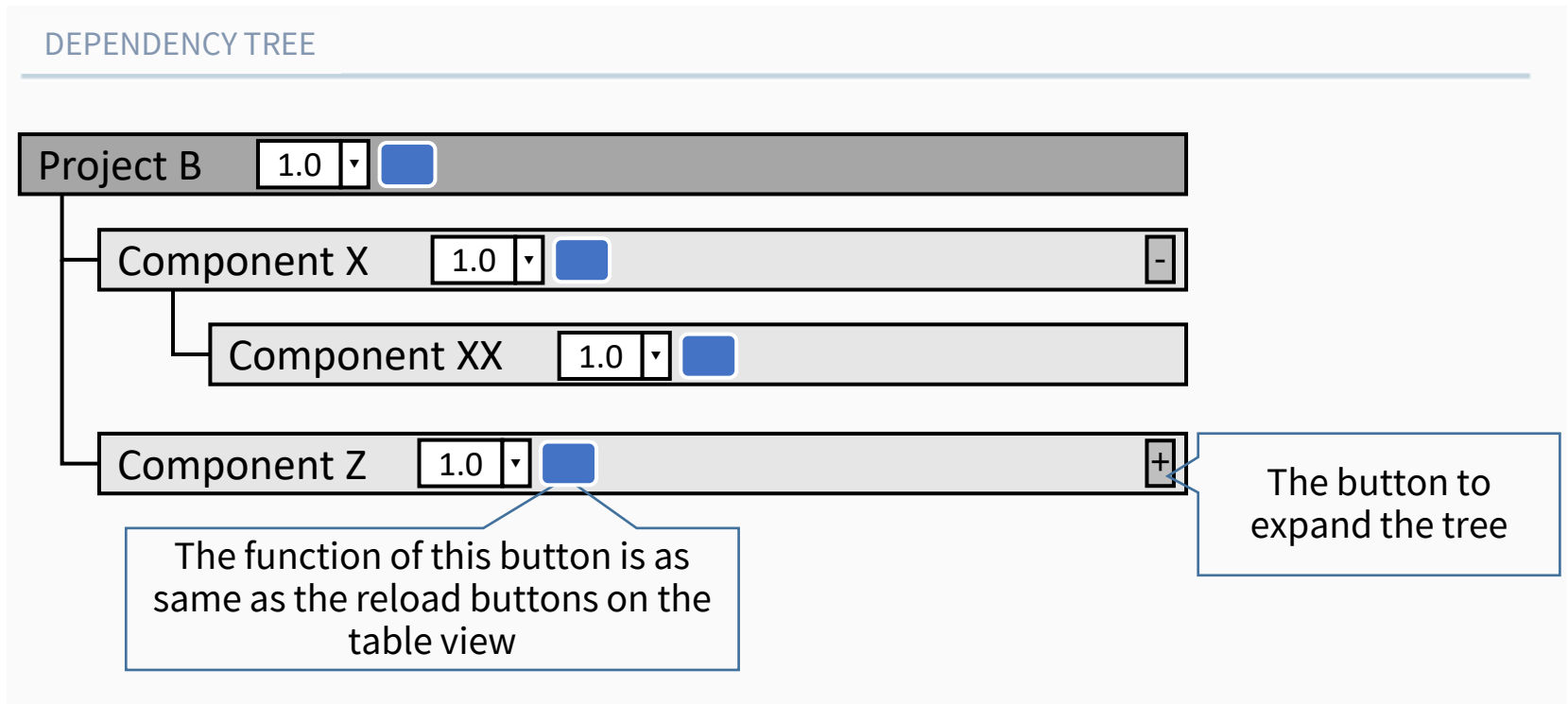
# Appendix: The Tree View of the Dependencies for this Proposal

Toshiba Corporation  
Shi QIU  
2022/05/19

# Appendix

- We will also implement a tree view of the dependencies for this proposal :

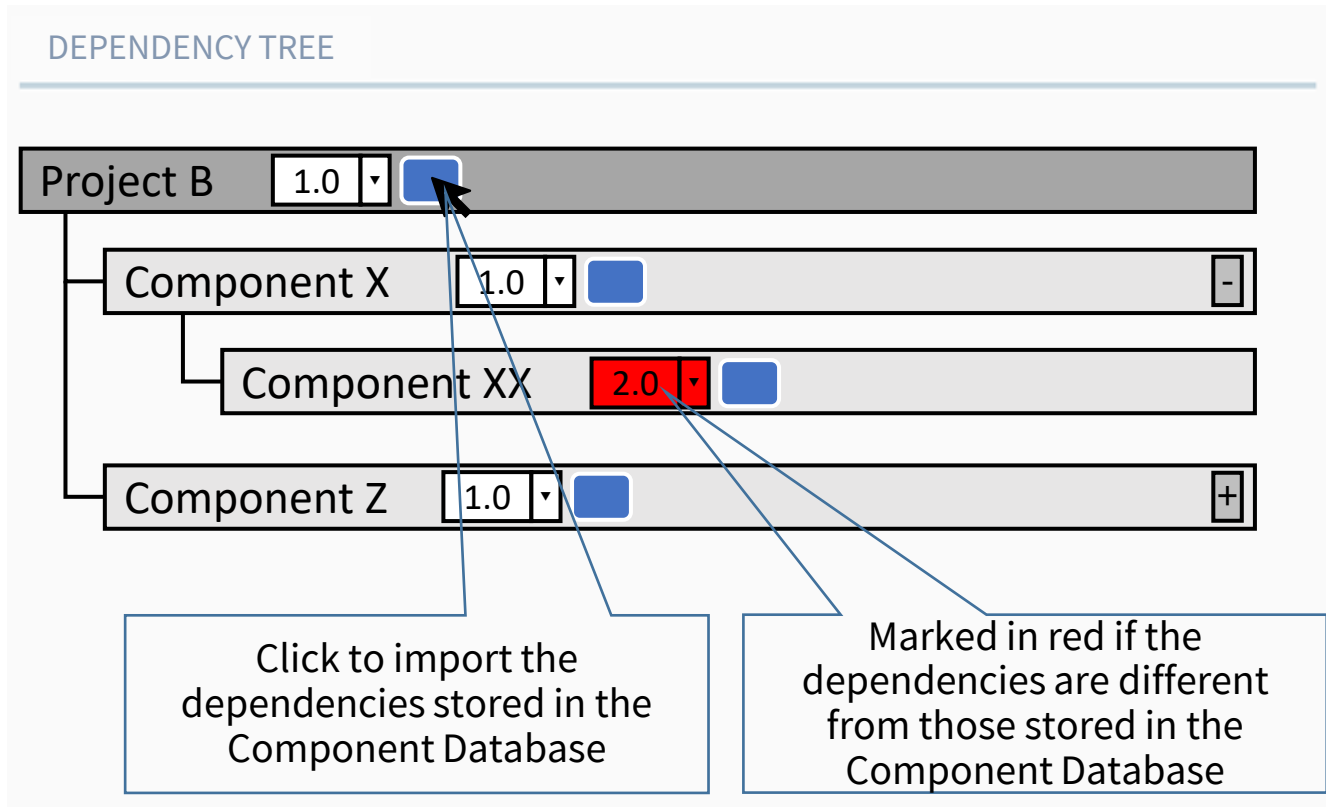
The direct and transitive dependencies can also be edited in this tree view.



# Appendix

- The reload button:

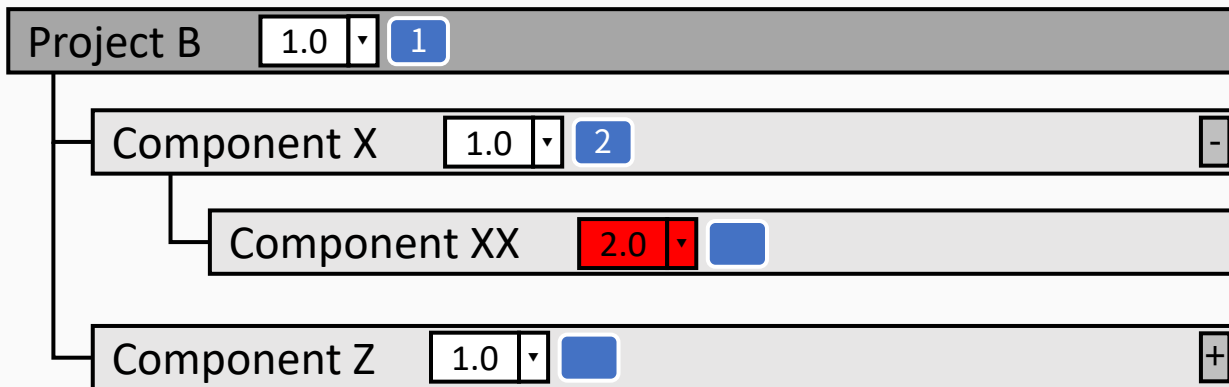
A button that allows the users to import the default dependencies stored in the component database.



# Appendix

- The influence range of the reload button:  
When clicking button 1: Import X, XX, Z dependencies  
When clicking button 2: Import XX dependencies

## DEPENDENCY TREE



**TOSHIBA**