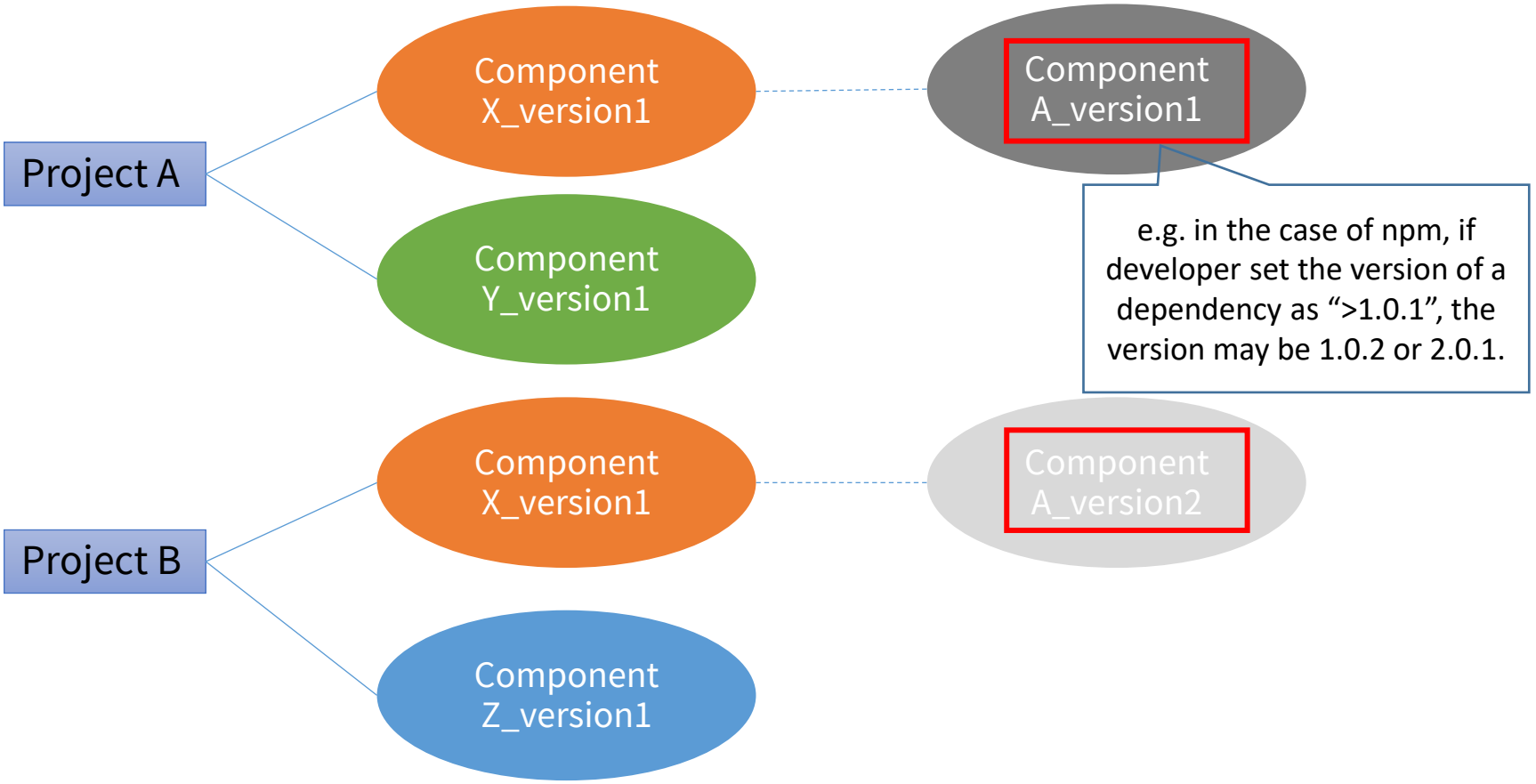
TOSHIBA

A New Dependency Management Function for SW360

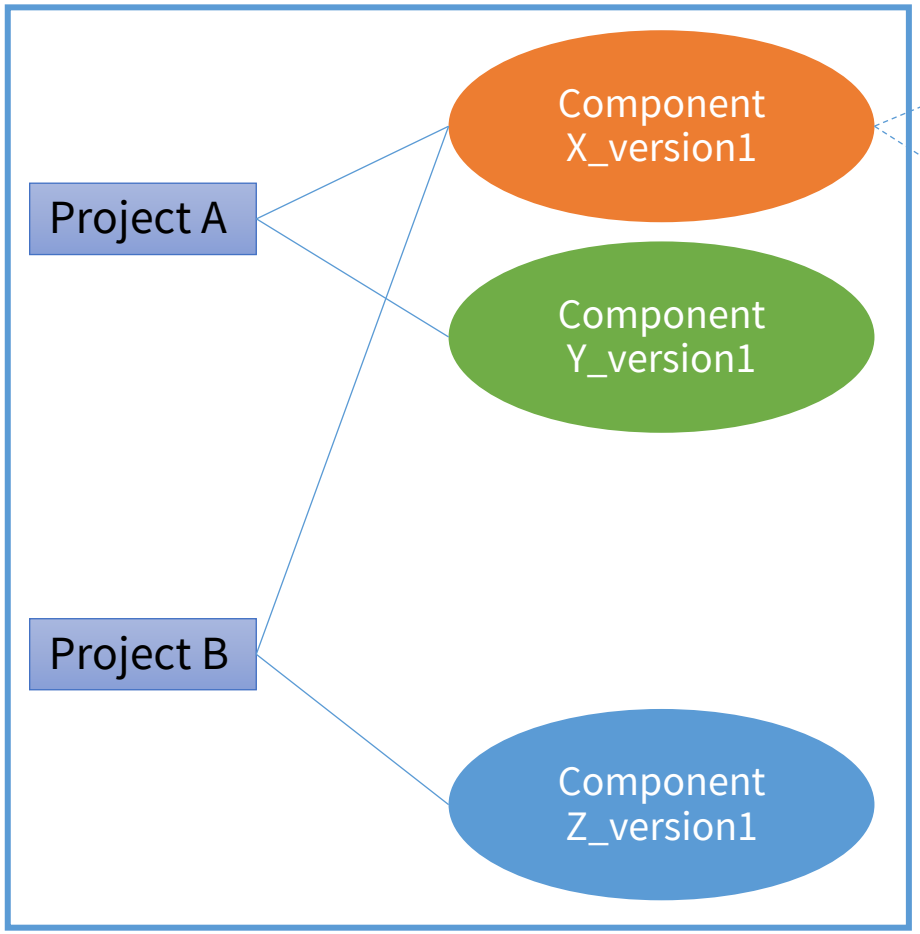
Problem

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

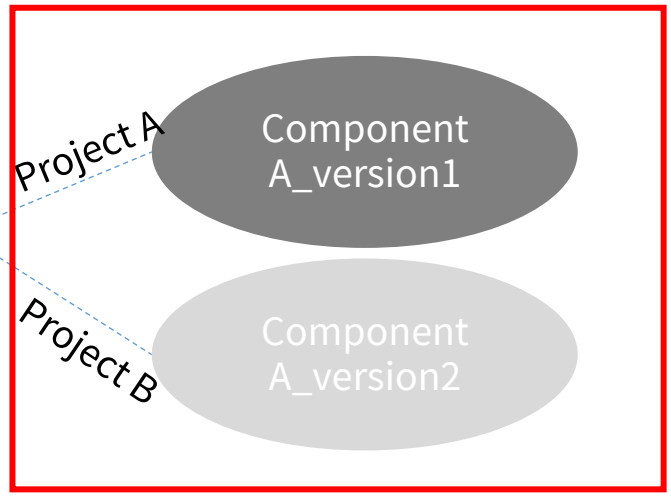


Problem

The current situation of SW360:



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

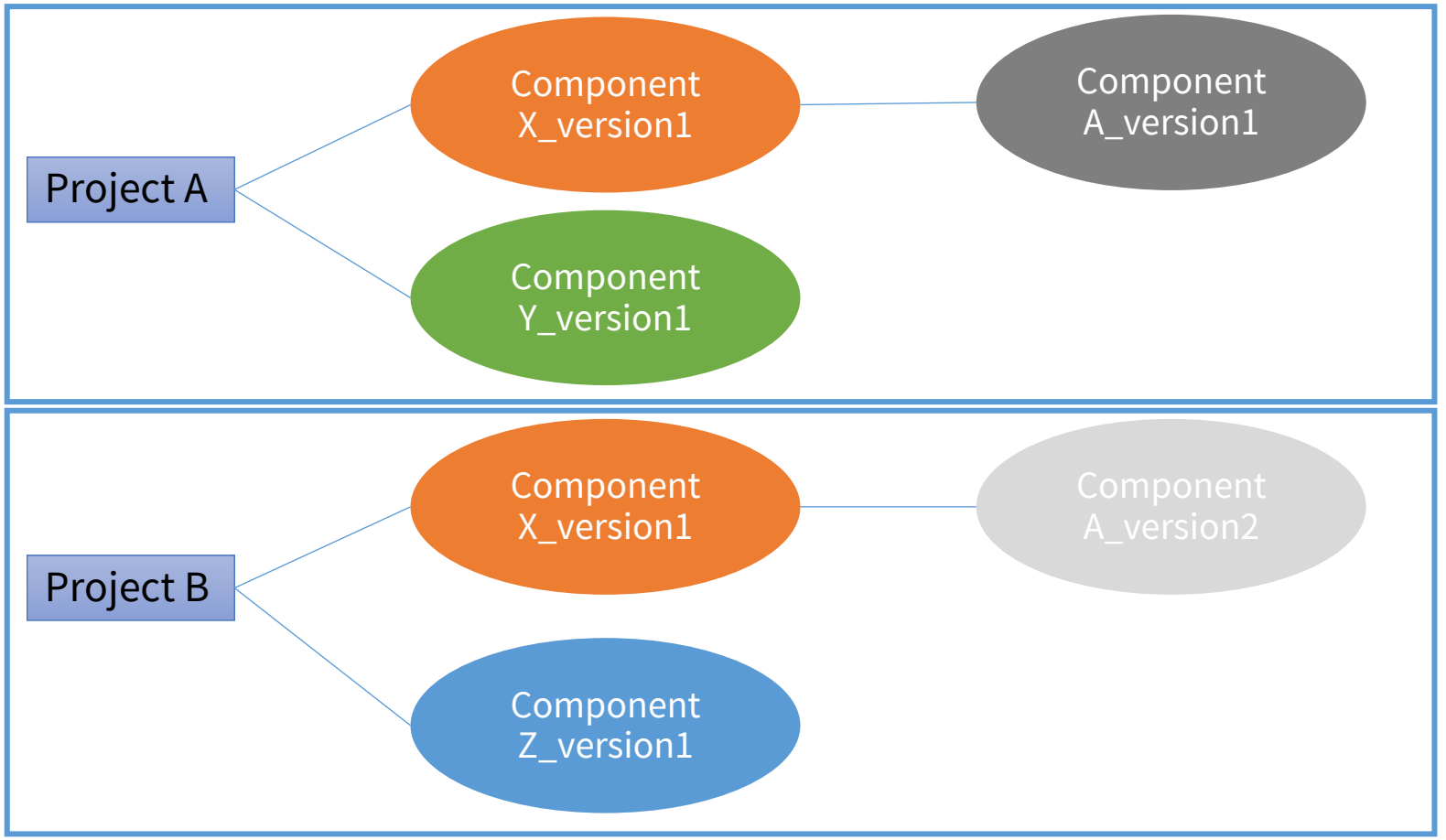


This case can not be registered in the current SW360

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

- Way of storing dependencies in project table (current)

Store only direct dependencies.


```
"releaseIdToUsage": {
  "369271c1d8284706958e023e30d4aa25": {
    "releaseRelation": "CONTAINED",
    "mainlineState": "OPEN",
    "comment": "",
    "createdOn": "2022-06-09",
    "createdBy": "admin@sw360.org"
  },
  "3adc12fa2eb94ec381526d91b988d581": {
    "releaseRelation": "UNKNOWN",
    "mainlineState": "OPEN",
    "comment": "",
    "createdOn": "2022-08-12",
    "createdBy": "admin@sw360.org"
  }
},
```


- Way of storing dependencies in project table (proposal):

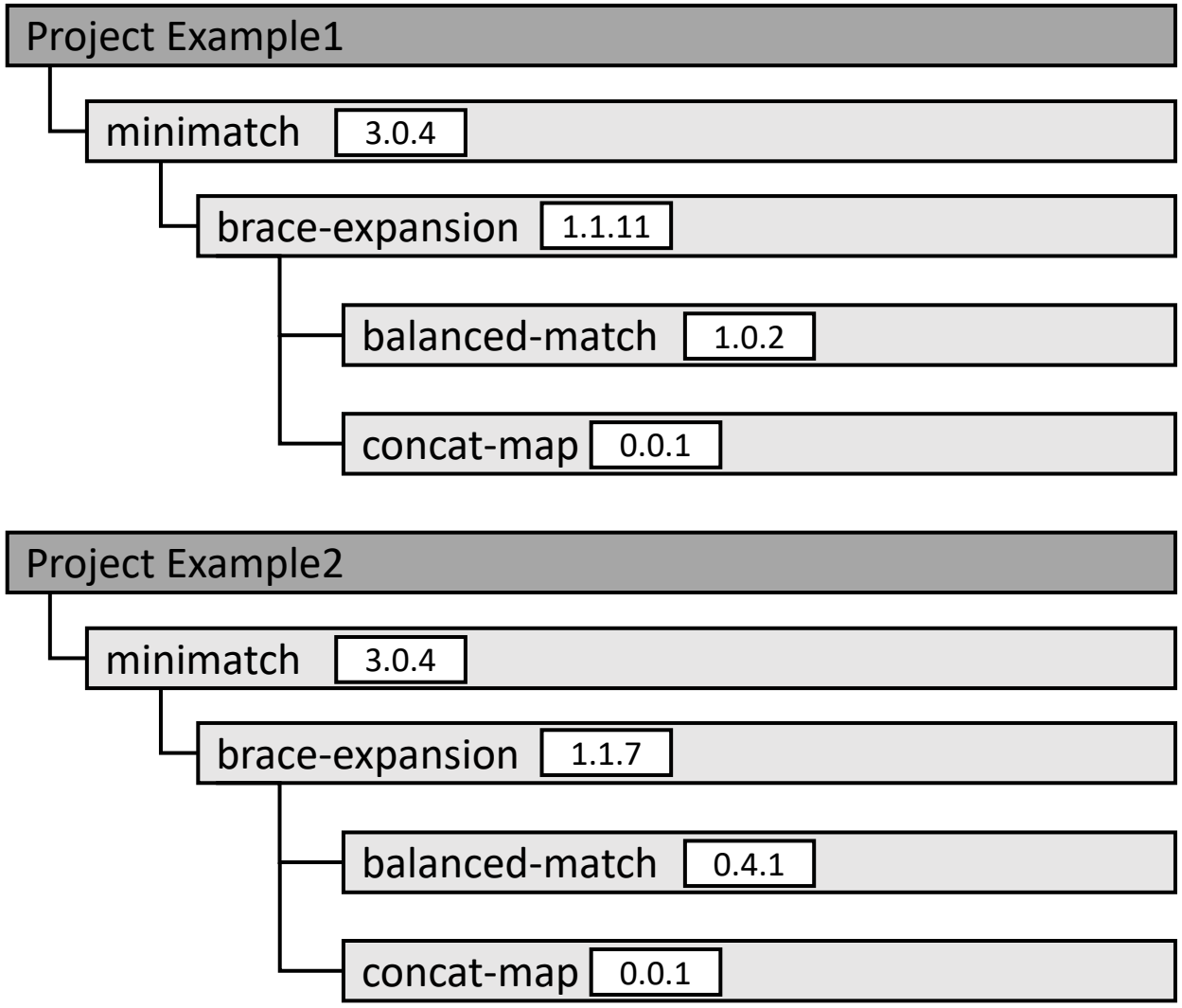

```
"createdBy": "admin@sw360.org",
"preevaluationDeadline": "",
"name": "abi-compliance-checker",
"roles": {},
"type": "project",
"createdOn": "2022-06-09",
"specialRisks3rdParty": "",
"releaseRelationNetwork": "[{\"comment\":\"\",\"releaseLink\":[],\"createBy\":\"admin@sw360.org\",\"createOn\":\"20
"externalUrls": {},
"remarksAdditionalRequirements": "",
"moderators": [],
```



```
[
  {
    "comment": "",
    "releaseLink": [],
    "createBy": "admin@sw360.org",
    "createOn": "2022-08-12",
    "mainlineState": "OPEN",
    "releaseId": "3adc12fa2eb94ec381526d91b988d581",
    "releaseRelationship": "UNKNOWN"
  },
  {
    "comment": "",
    "releaseLink": [
      {
        "comment": "",
        "releaseLink": [],
        "createBy": "admin@sw360.org",
        "createOn": "2022-06-09",
        "mainlineState": "OPEN",
        "releaseId": "489391974 68b45e18fcbf5b4cca7636a",
        "releaseRelationship": "CONTAINED"
      }
    ],
    "createBy": "admin@sw360.org",
    "createOn": "2022-06-09",
    "mainlineState": "OPEN",
    "releaseId": "369271c1d8284706958e023e30d4aa25",
    "releaseRelationship": "CONTAINED"
  }
]
```


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"

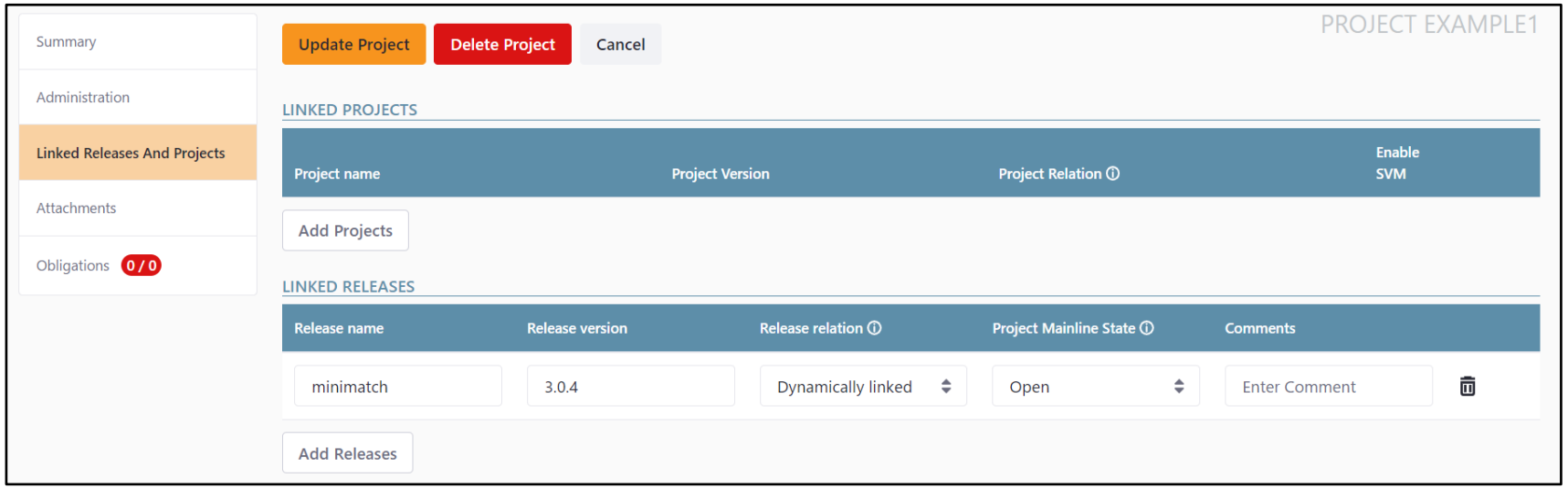
},

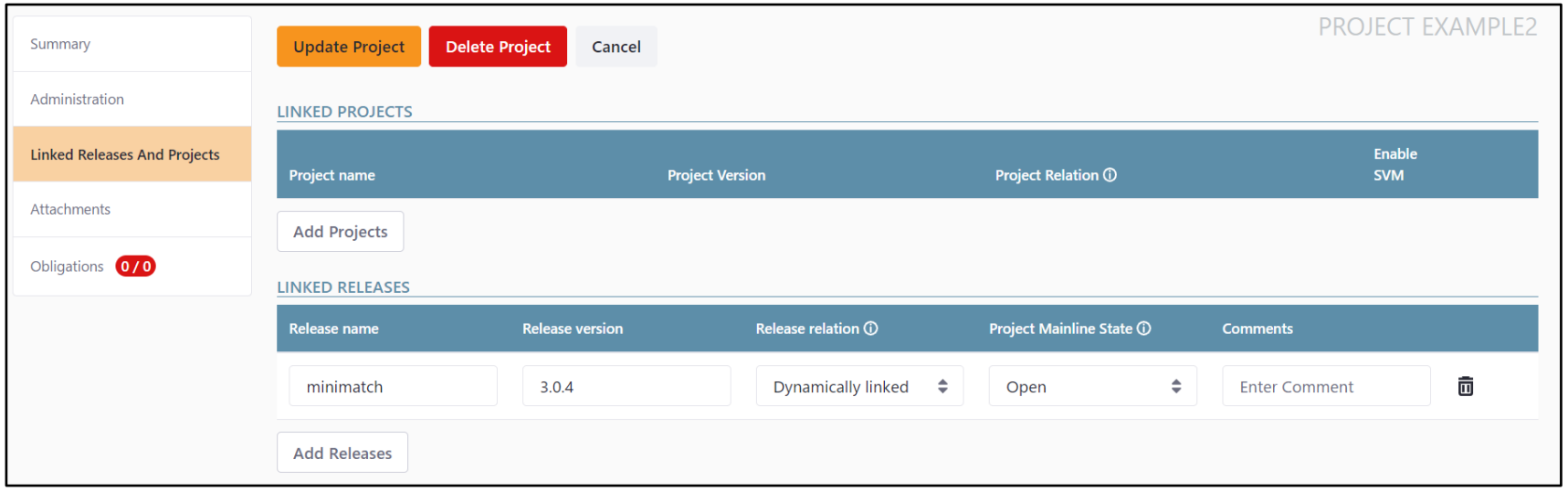
****

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

Example

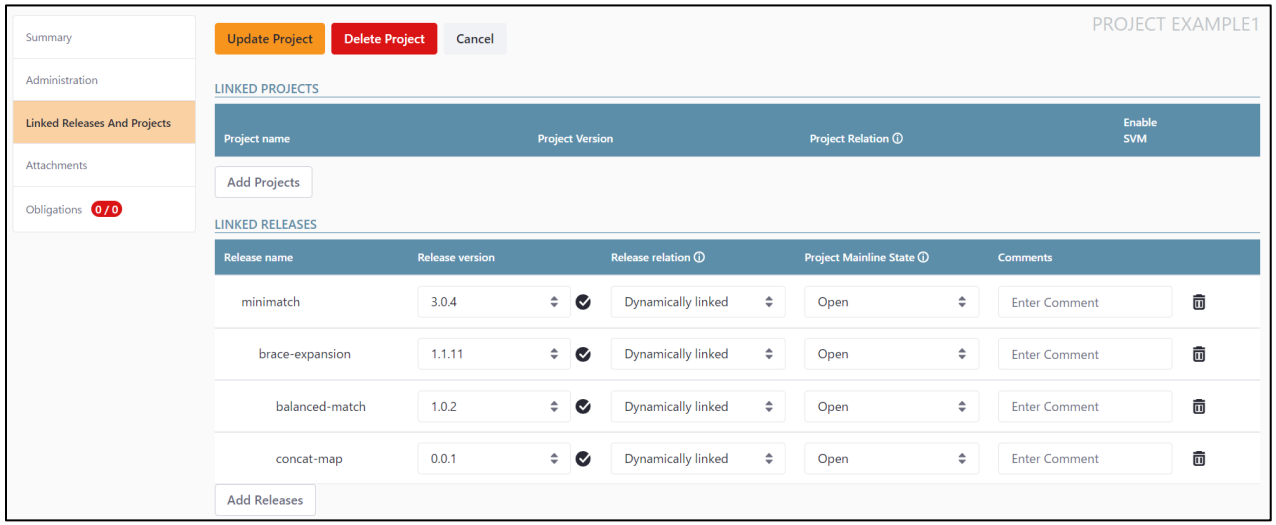
- The current situation:

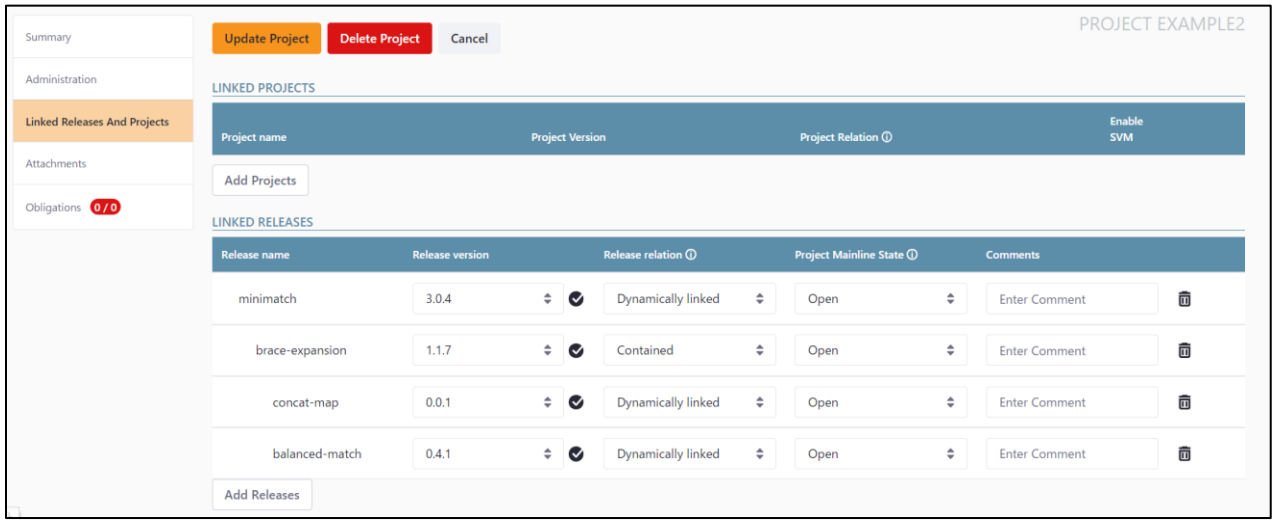




Example

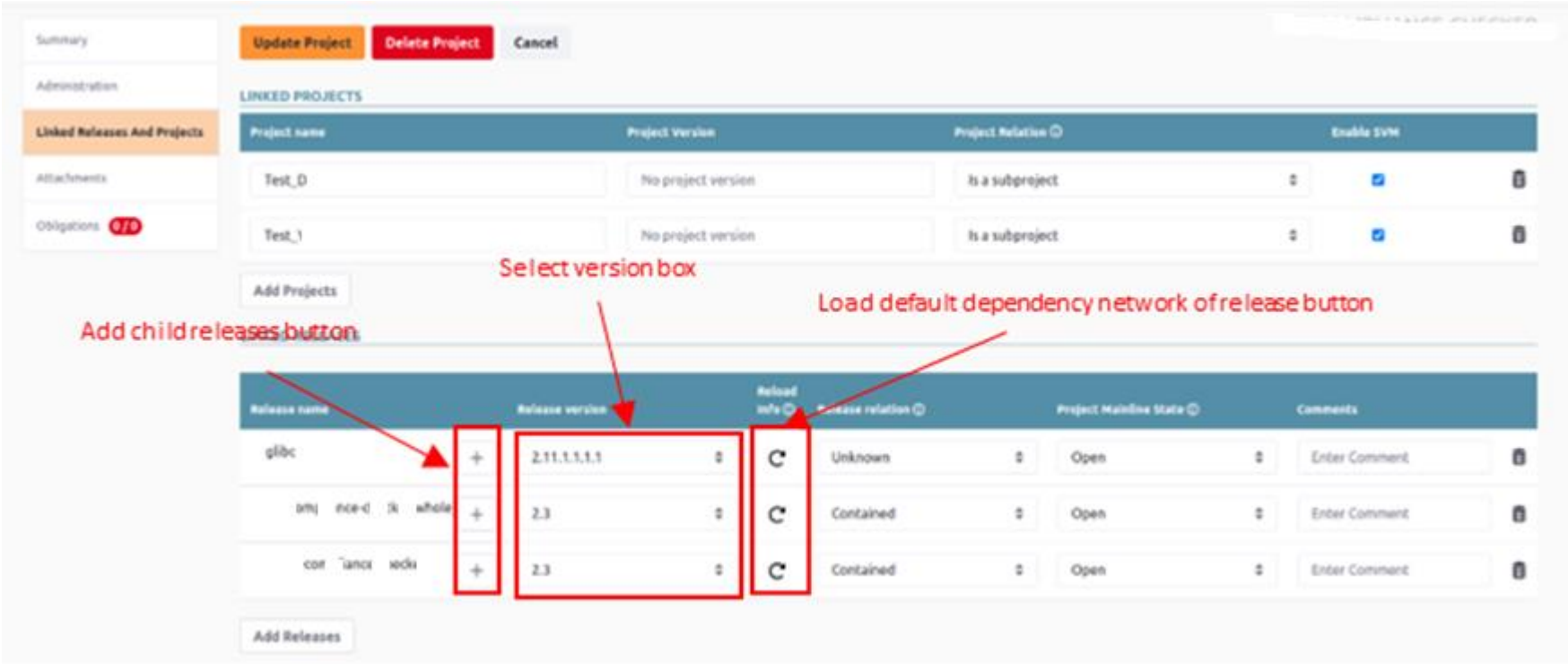
- The proposal:





GUI

- Change the GUI of the **Linked Releases And Projects** tab (at **Edit project** page)

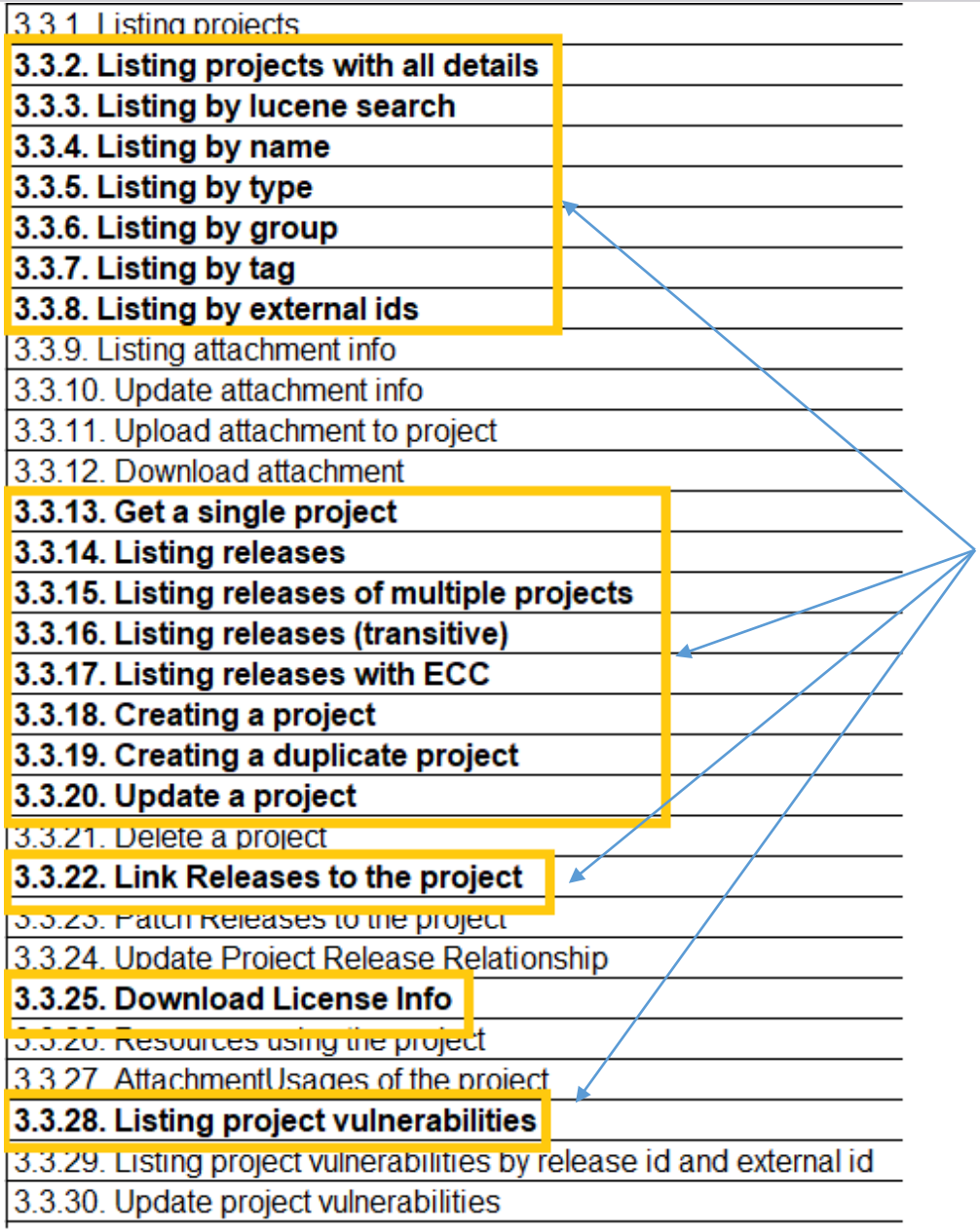


GUI

- Keep the GUI of related functions, update the logic in each related function to make it compatible with new Dependency Network features.

- Related functions
  - License Clearing tag
  - Clearing Request
  - ECC tag
  - Rest API function
  - Import/Export function
  - Change Log function
  - SPDX import

API



- **API: Updated**

API

Project REST API
3.3.1. Listing projects
3.3.2. Listing projects with all details
3.3.3. Listing by lucene search
3.3.4. Listing by name
3.3.5. Listing by type
3.3.6. Listing by group
3.3.7. Listing by tag
3.3.8. Listing by external ids
3.3.9. Listing attachment info
3.3.10. Update attachment info
3.3.11. Upload attachment to project
3.3.12. Download attachment
3.3.13. Get a single project
3.3.14. Listing releases
3.3.15. Listing releases of multiple projects
3.3.16. Listing releases (transitive)
3.3.17. Listing releases with ECC
3.3.18. Creating a project
3.3.19. Creating a duplicate project
3.3.20. Update a project
3.3.21. Delete a project
3.3.22. Link Releases to the project
3.3.23. Patch Releases to the project
3.3.24. Update Project Release Relationship
3.3.25. Download License info
3.3.26. Resources using the project
3.3.27. AttachmentUsages of the project
3.3.28. Listing project vulnerabilities
3.3.29. Listing project vulnerabilities by release id and external id
3.3.30. Update project vulnerabilities

•API: Removed.

•In case, you would like to:
- **Patch Releases to the project**
- **Update Project Release Relationship**

•Please use the "3.3.20: Update a project" API instead.

Point

- Why this feature?
  - The dependency information of the project will be changed if the information on the component page is changed. The project owner will not be notified about this.
  - We can not export the correct and stable SBOM since the dependency network is created dynamically. It varies from time to time.
  - Separate the responsibility for the project and the one for the component is important for “InnerSource”.

Point

- All changes are only done on the “Project” page, the “Component” page will not change anything.

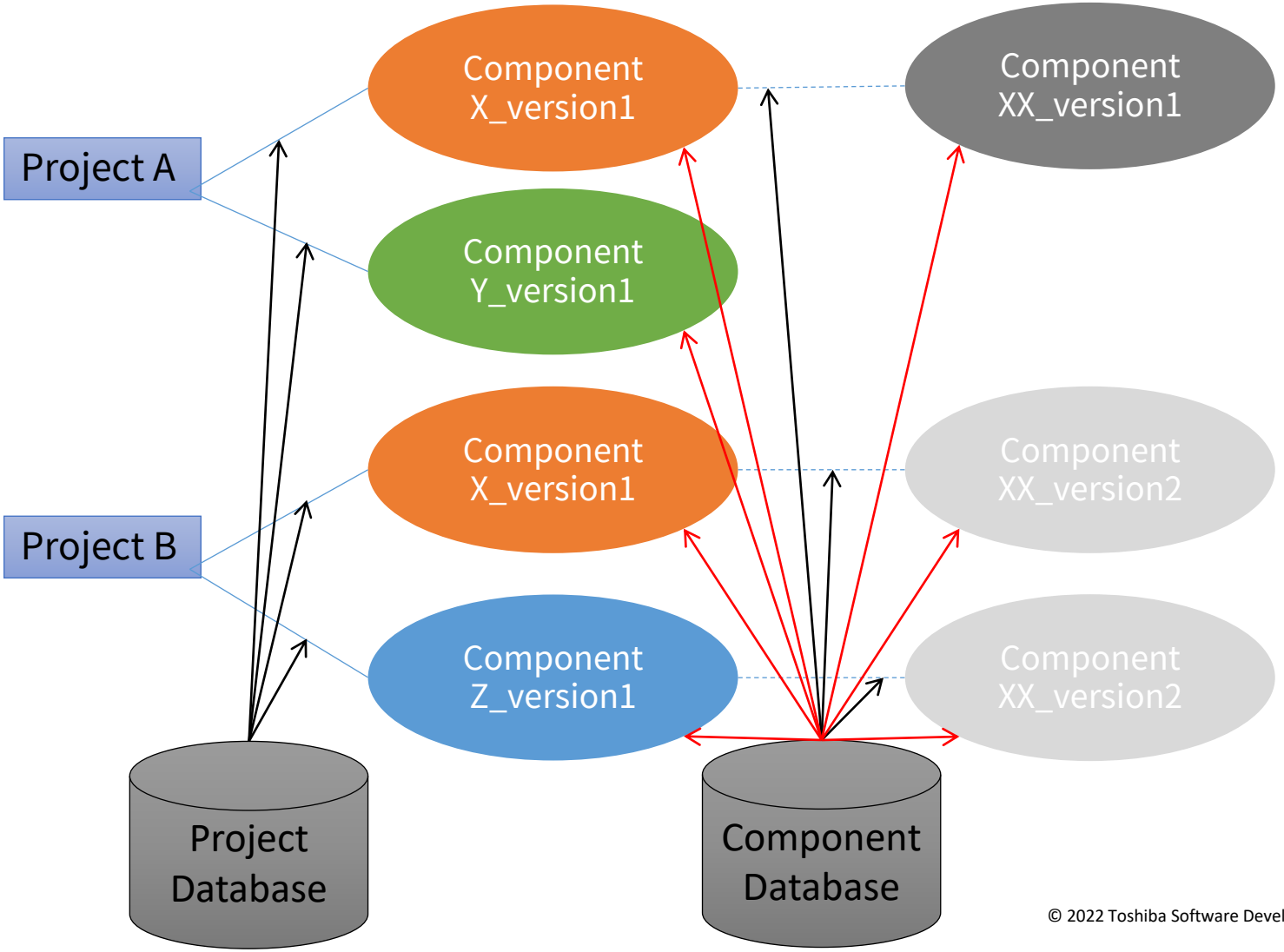
- The dependency information in the “Component” page will be seen as the place storing the “default” information. It will keep the same with the latest information in the ecosystem.

- The dependency information in the “Project” page will be seen as the place storing the “actual” information. It will not change even if the “default” information is updated. But we will let the owner know which dependency of his project is outdated (which means different from the one on the “Component” page).

Proposal

Black arrow: Dependency information
Red arrow: Component field information

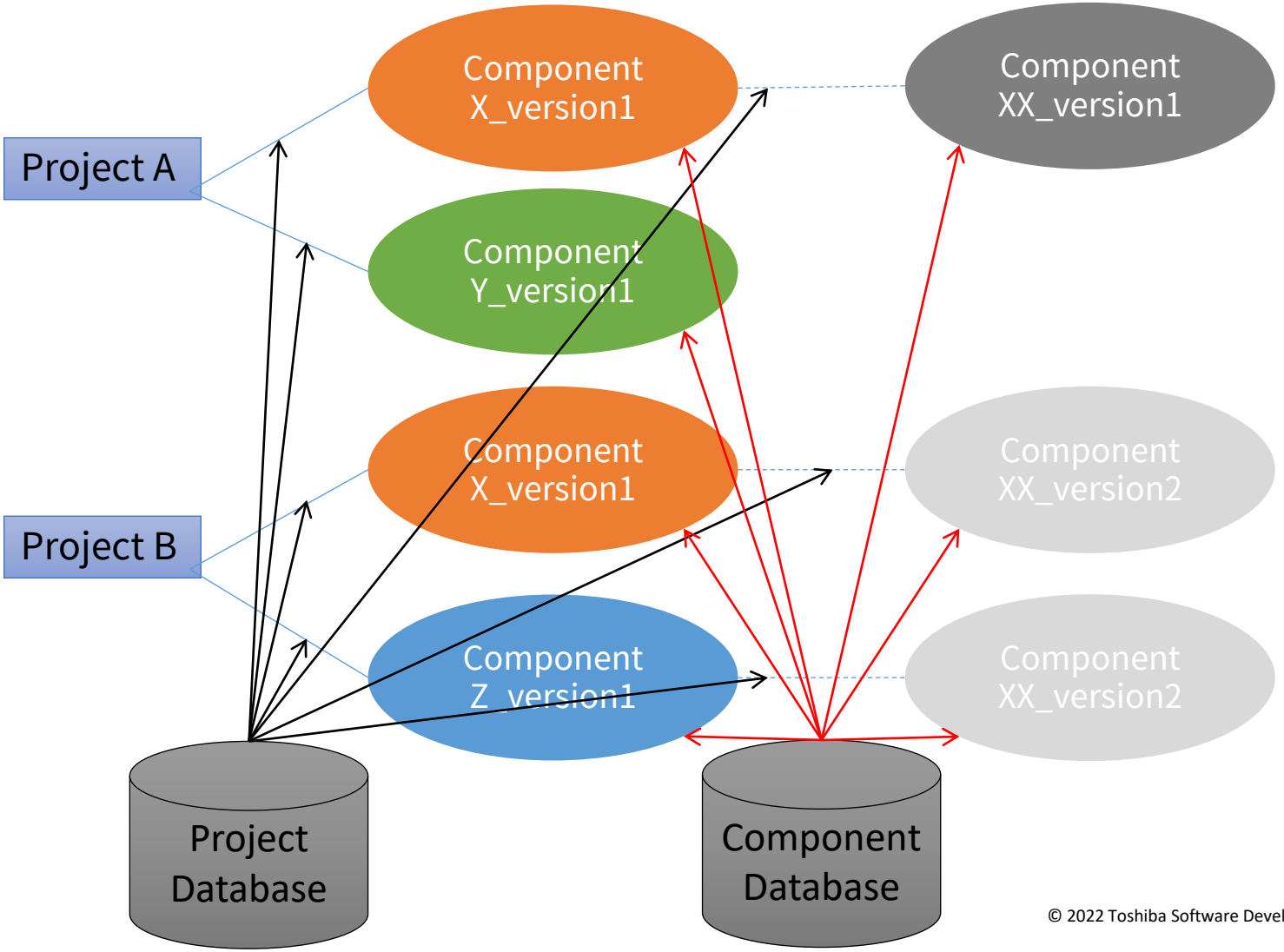
- Way of storing dependencies (current)



Proposal

- Way of storing dependencies (proposal)

Black arrow: Dependency information
Red arrow: Component field information



TOSHIBA