

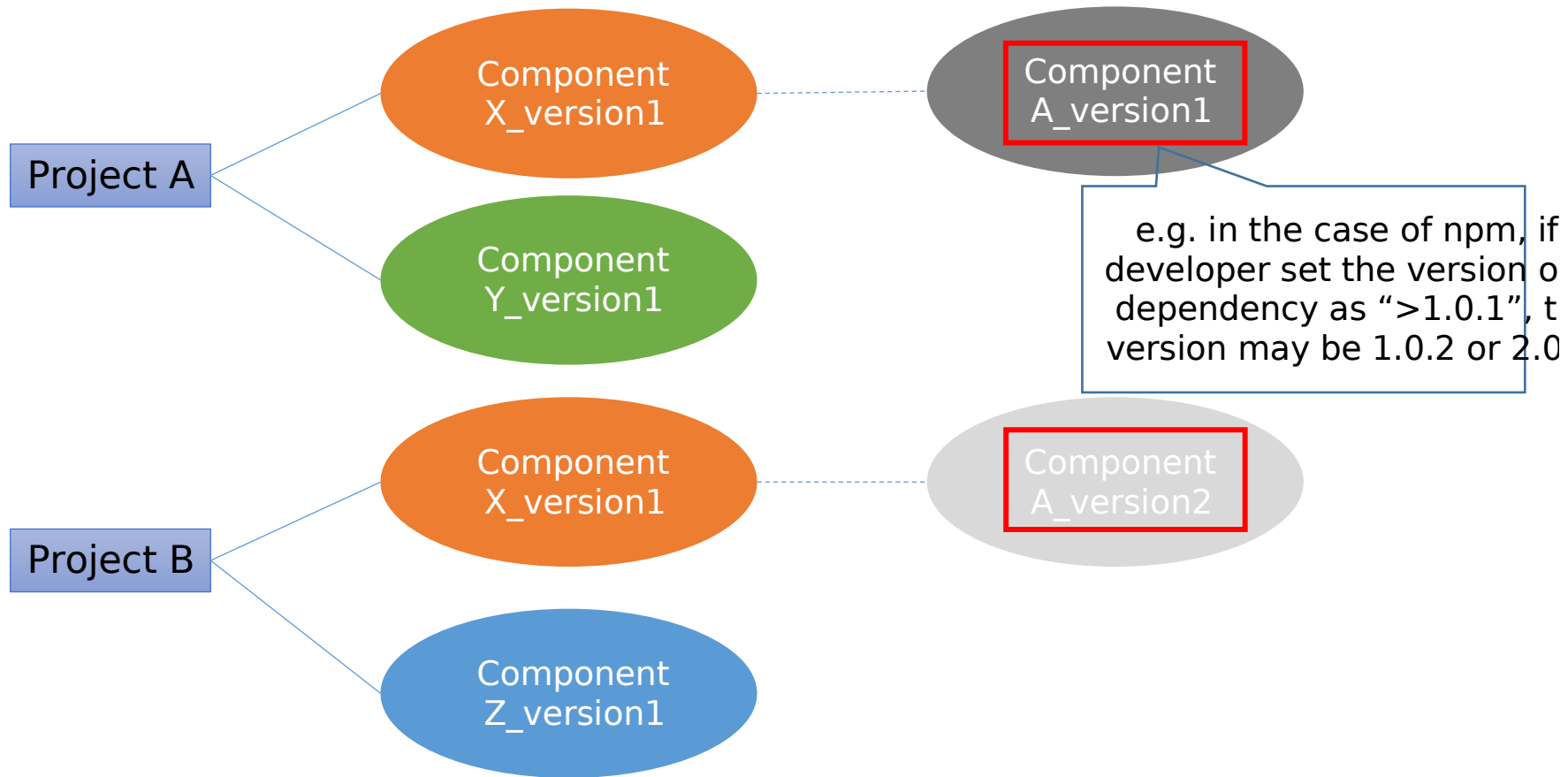
TOSHIBA

A New Dependency Management Function for SW360



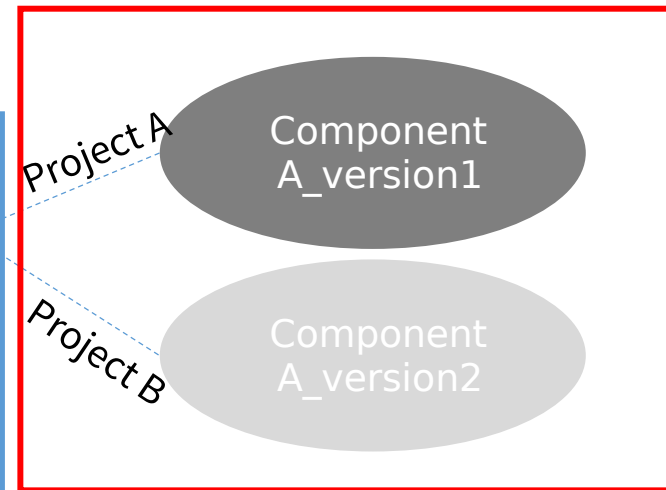
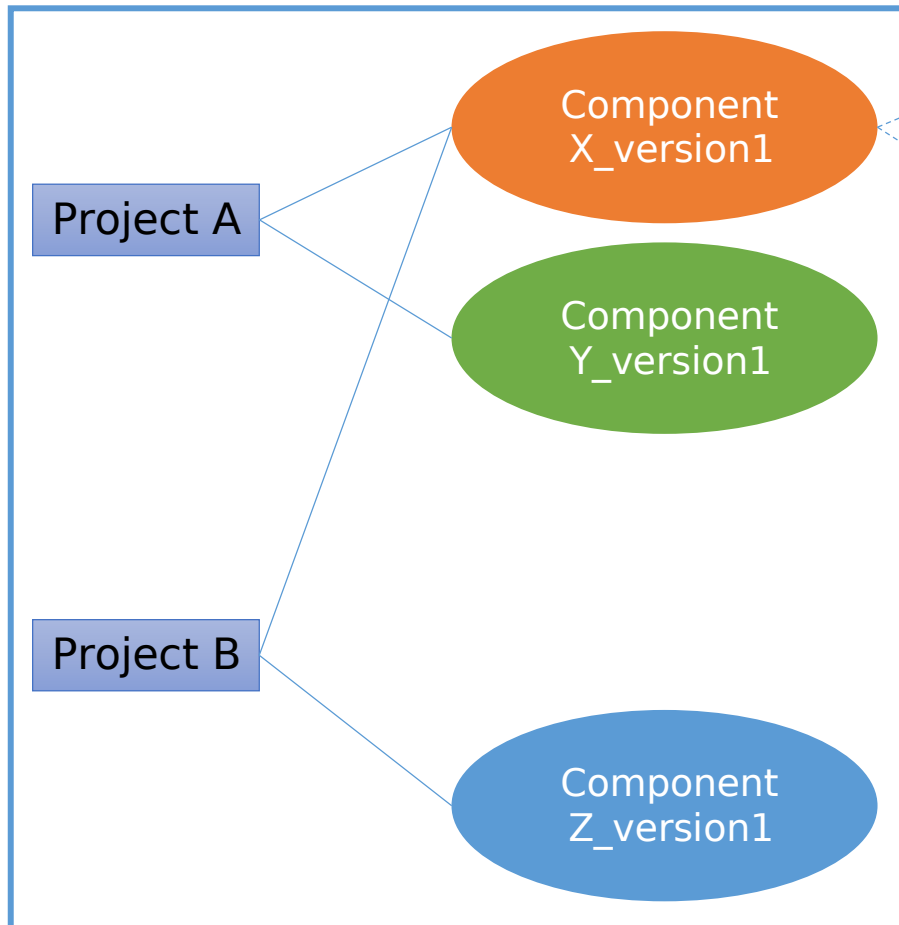
Problem

- 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:



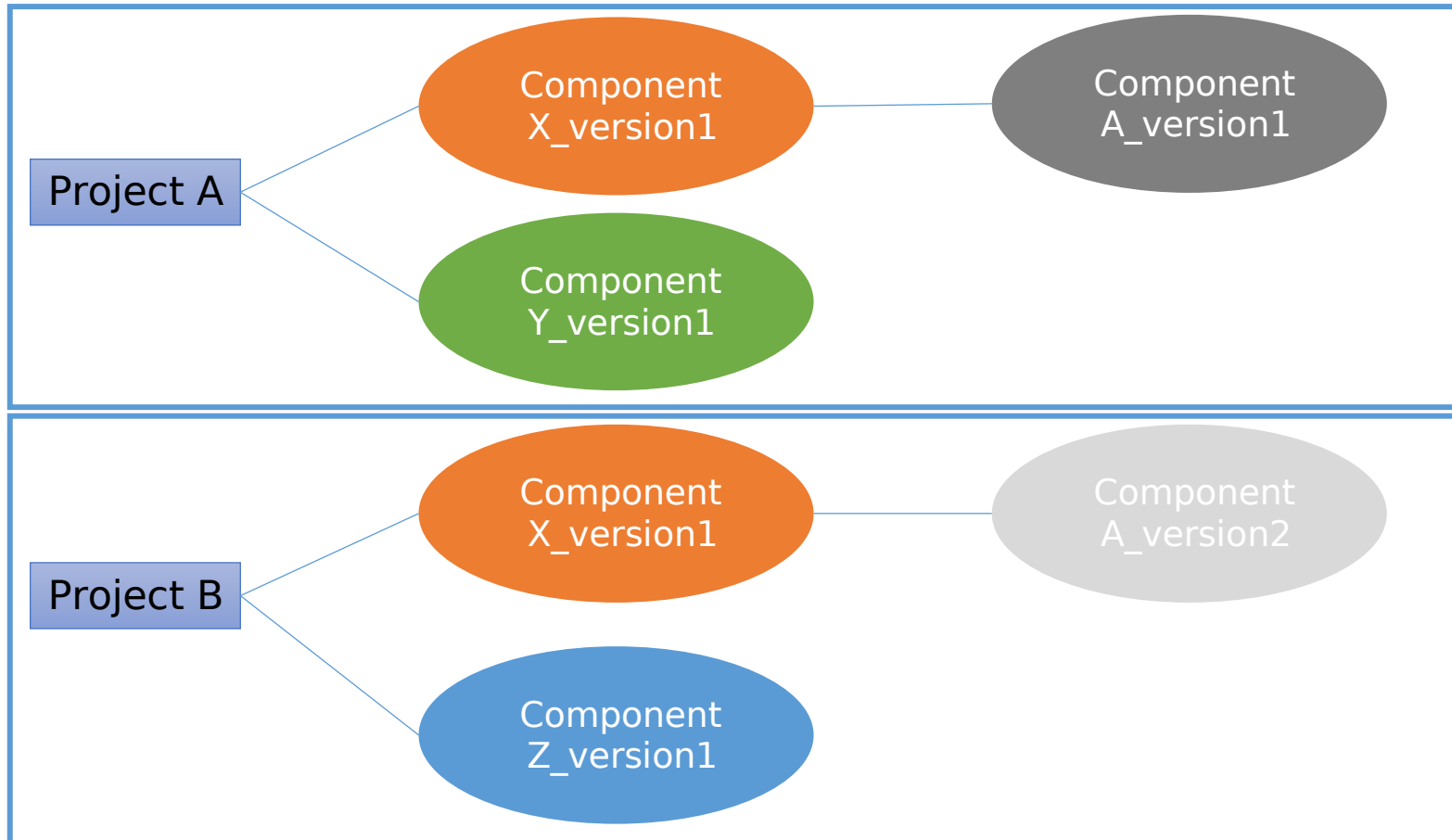
This case can not be registered in the current SW360

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

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"
  }
},
```

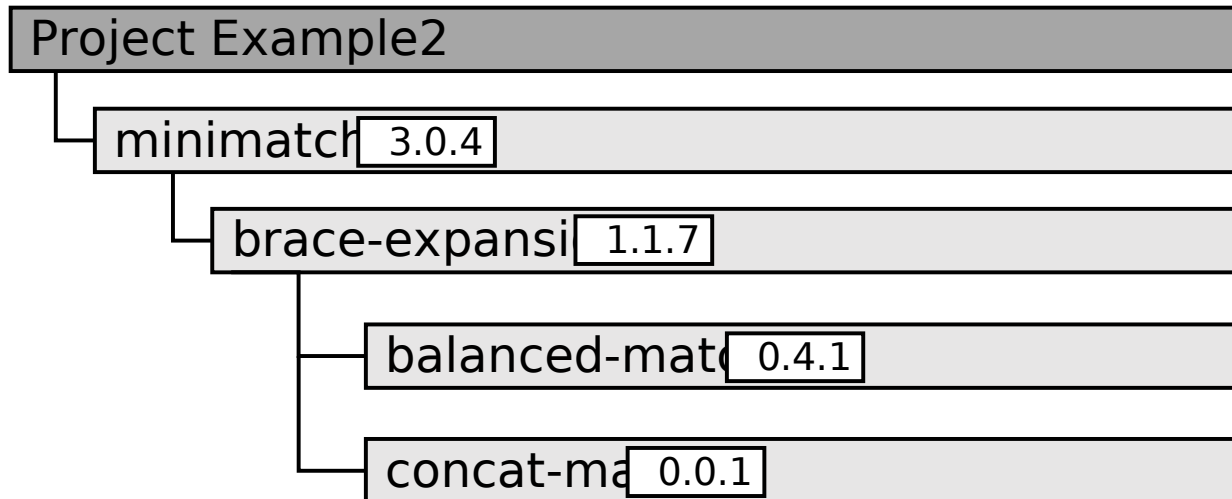
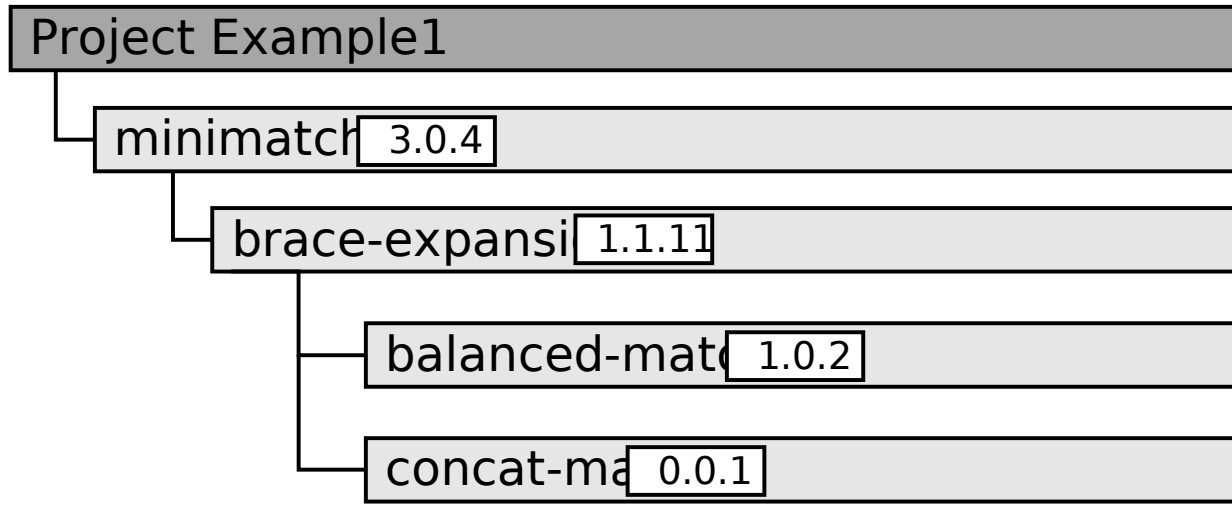
- Way of storing dependencies in project table (proposal):

```
"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":"2022-06-09","releaseId":"48939197468b45e18fcbf5b4cca7636a","releaseRelationship":"CONTAINED"}]",
"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": "48939197468b45e18fcbf5b4cca7636a",
      "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:

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 

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 

Add Releases

Example

- The proposal:

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
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 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
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

Point

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

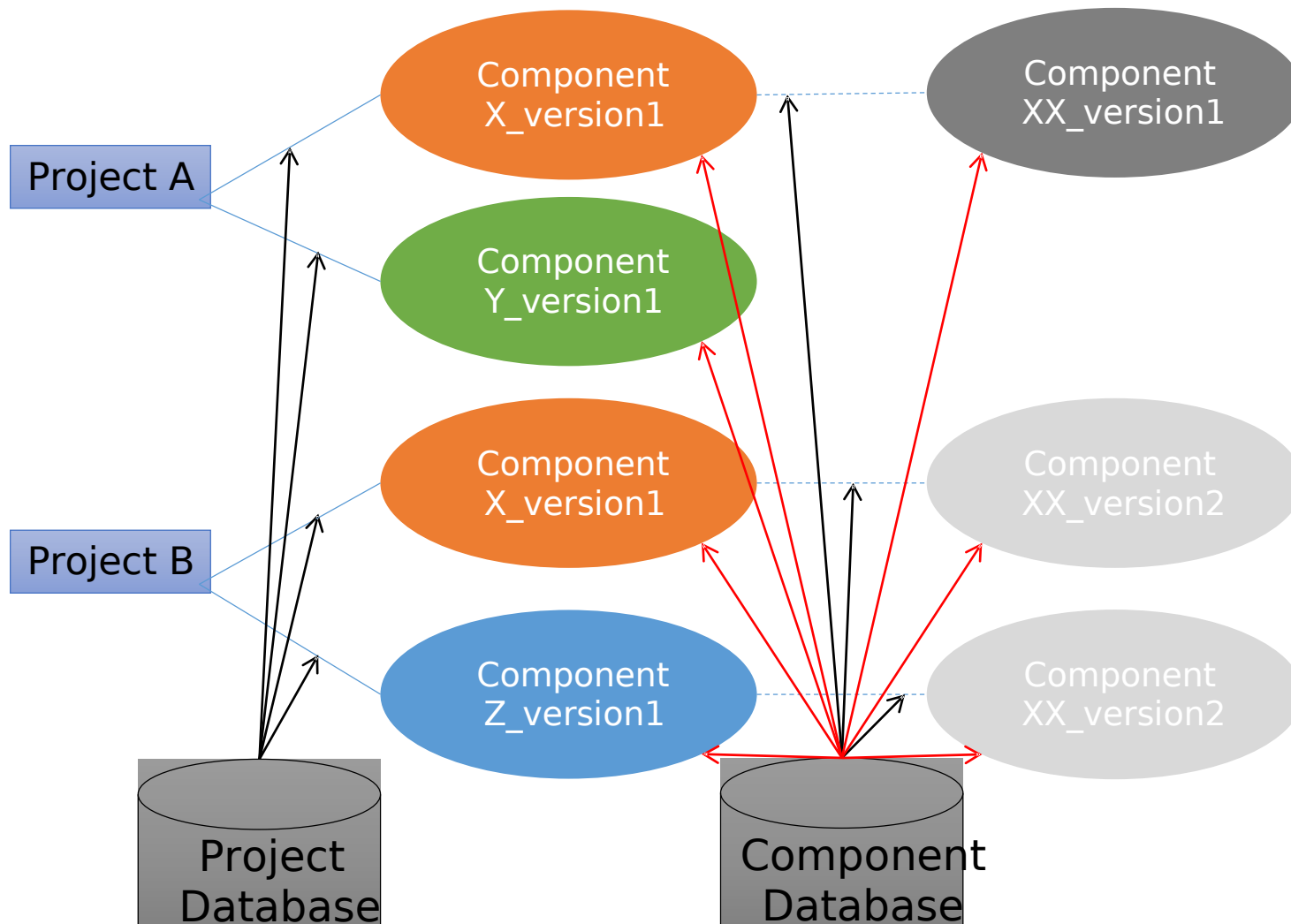
Point

- All changes are only done on the “Project” page, the “Component” page will not change anything.
- The dependency information in the “Component” page can be seen as the place storing the “default” information. It will keep the same with the latest information in the “Project” page.
- The dependency information in the “Project” page can be seen as the place storing the “actual” information. It will change even if the “default” information is updated. This 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

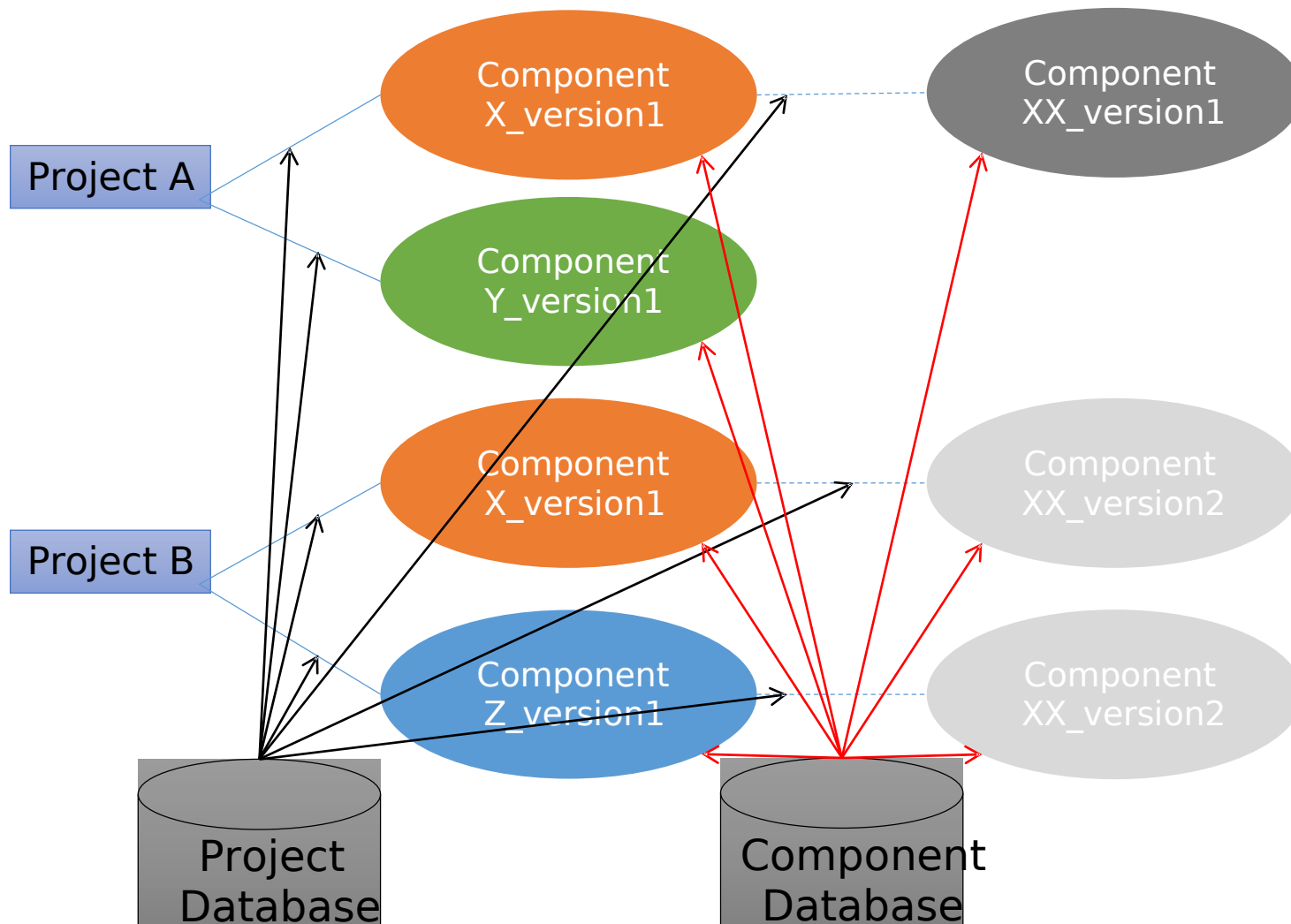
- Way of storing dependencies (current)



Proposal

Black arrow:
Dependency information
Red arrow: Component
field information

- Way of storing dependencies (proposal)



TOSHIBA