HZB Helmholtz Zentrum Berlin

SUMO

An EPICS SUpport MOdule Manager

Overview

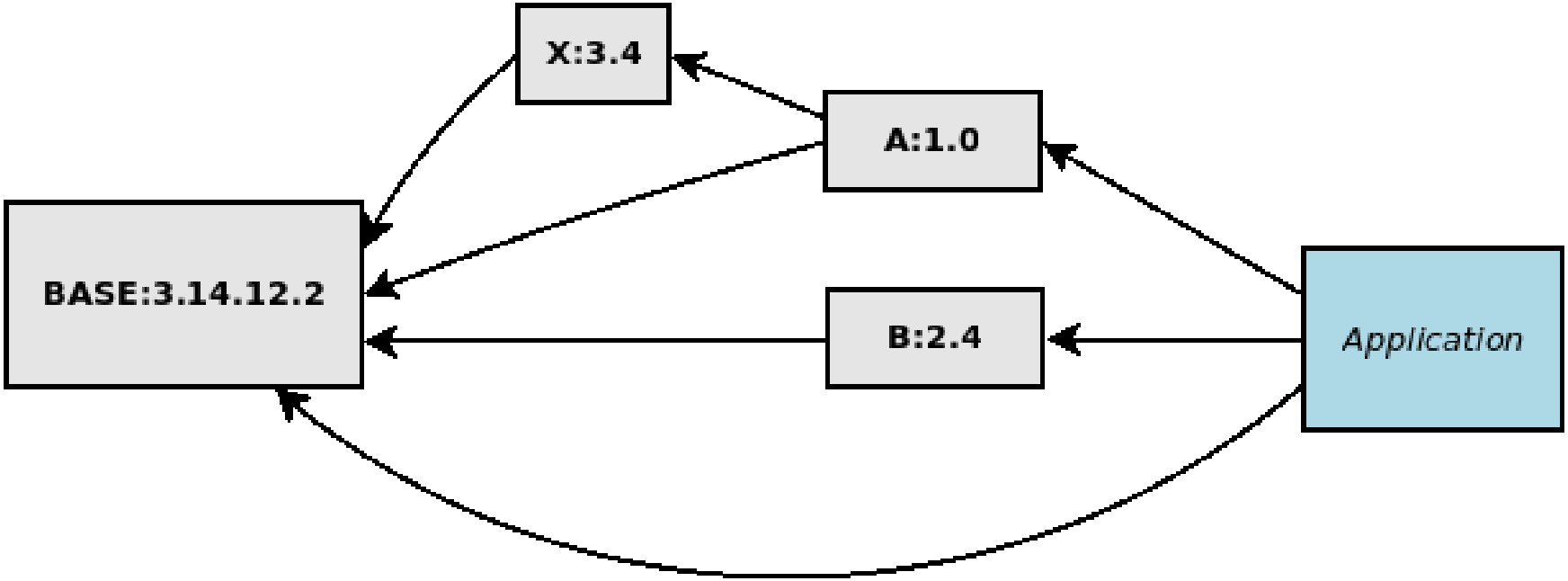
- EPICS Support Modules
- The problem with support module dependencies
- The solution: SUMO Builds
- How SUMO works
- How to use SUMO
- How to install SUMO
- Links and references

EPICS Support Modules

- Types
  - Support for specialized hardware (device support)
  - New record types
  - Specialized panels
  - Everything your application needs that is not part of EPICS Base
- Dependency definition
  - Basically „make“ macros in configure/RELEASE
  - RELEASE file must be edited by hand for the *actual* support module paths.

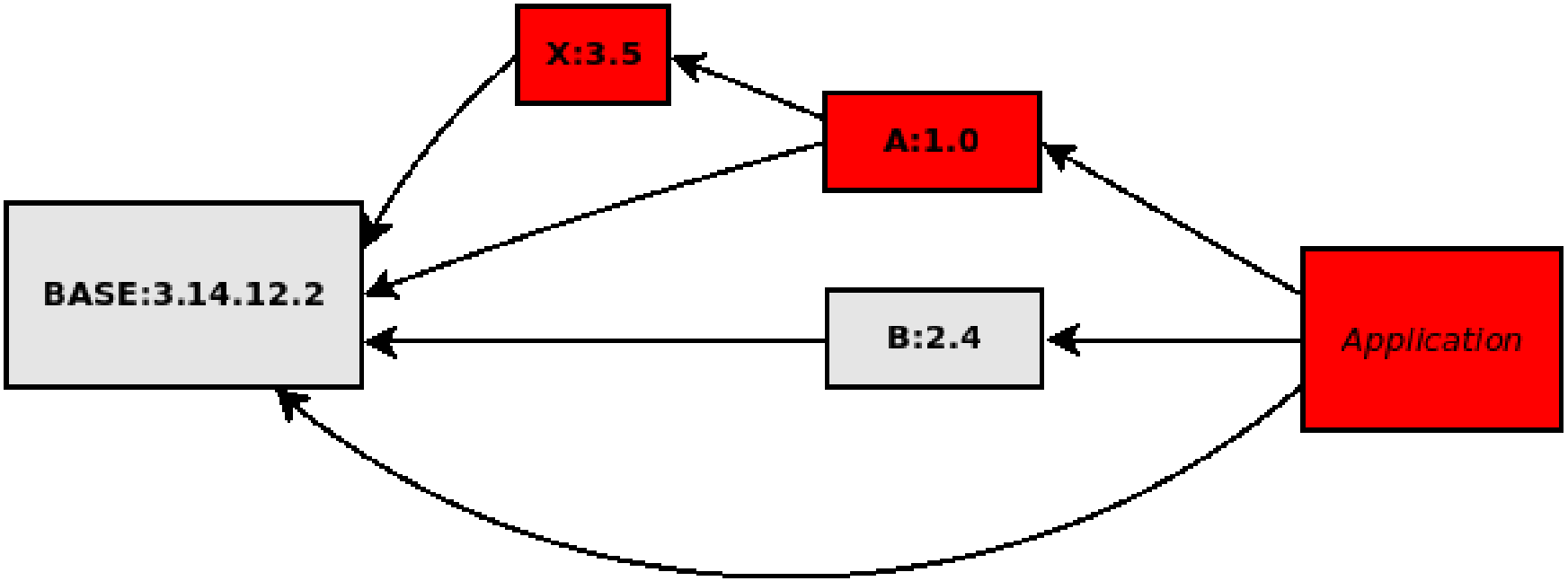
The problem with support module dependencies

Example: An EPICS Base, three Supports A, B and X and an Application:



The problem with support module dependencies

Change of support X requires recompilation of A and the application, although X may not be mentioned in it's RELEASE file

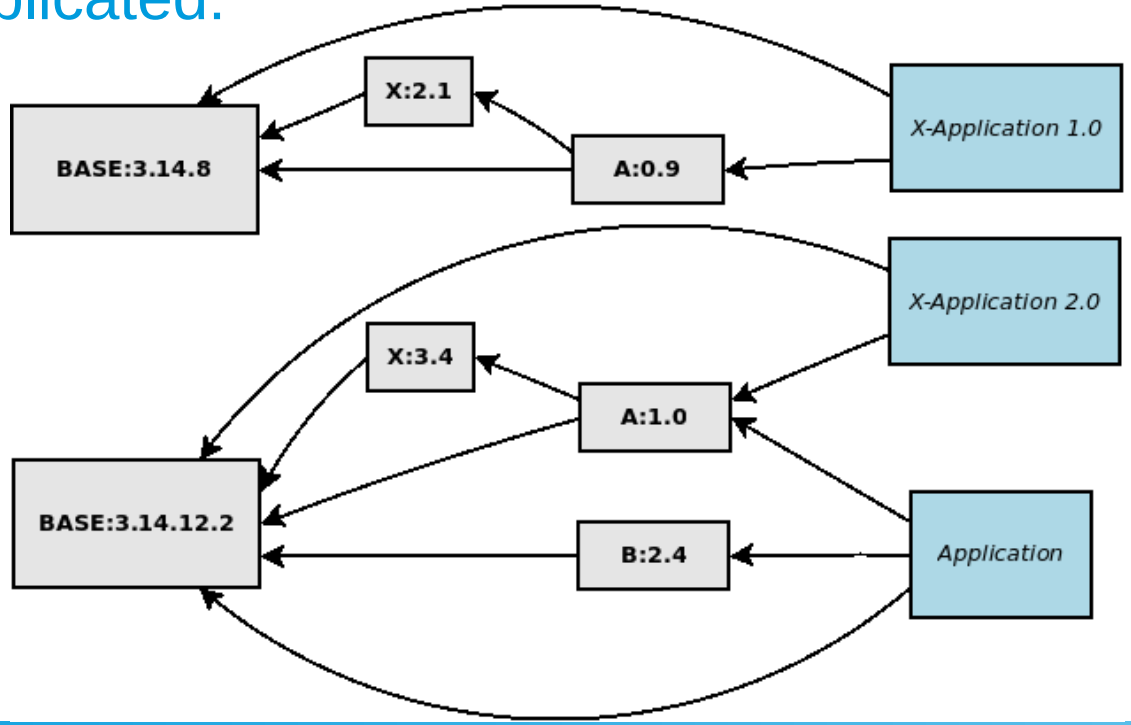


The problem with support module dependencies

- How to find all modules that use module X ? You must do a text search of RELEASE files in all other modules.
- How to handle the new compiled version of module A ? The sourcecode didn’t change but it uses a different version of X and may behave differently. Do you give it a new version number although it’s source hasn’t changed ?
- How do you know which version of X your application uses ? Since it doesn’t use X directly, this module is not mentioned in file RELEASE.

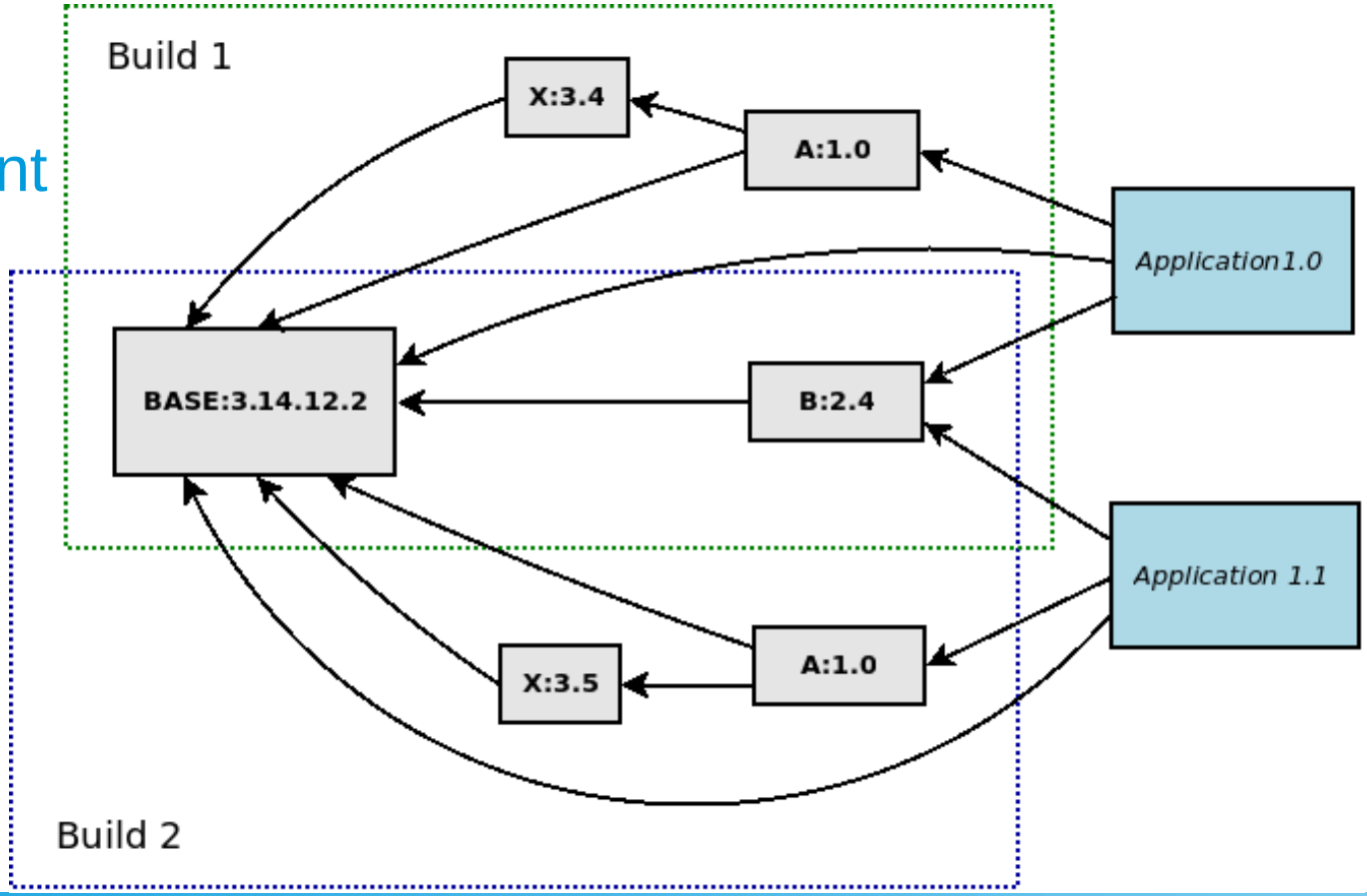
The problem with support module dependencies

In a typical installation with many applications the situation gets more complicated:



The solution: SUMO Builds

Supports can be re-used in consistent build sets, „builds“



How SUMO works – Configuration

- The configuration file(s) is always named „sumo.config“. It is in JSON file format.
- „sumo.config“ is searched at different locations and the results are merged
- Your application or EPICS support usually has it’s list of needed supports in file „configure/MODULES“. With this file, you do not need to specify the list of supports on the command line. This file has a complete set of the supports and their version that your application requires.

How SUMO works – MODULES file

Example of a MODULES file

- „module“ has support modules and their version
- Optional „alias“ entries define the name of the macro definition in the generated RELEASE file

```
{
    "alias": [
        "BASE:EPICS_BASE",
        "BESSYRULES:BESSY_RULES",
        "SEQ:SNCSEQ",
        ...
    ],
    "module": [
        "BASE:R3-14-12-7-bessy5",
        "BESSYRULES:R2-21",
        "SEQ:R2-2-3",
        ...
    ]
}
```

How SUMO works – Steps to create a new build

- Checks if given list of support modules is complete
- Determines which support modules need to be compiled and which can be re-used from other existing builds
- Generates a Makefile for build
- Checks out support modules from VCS
- Generates RELEASE files in support modules
- Calls „make“ on generated build makefile

How SUMO works – Data management

- Data on support module versions, VCS access data and dependencies is held in JSON file „DEPS.DB“
- Data on builds, the support modules they contain and support modules re-used from other builds are held in JSON file „BUILDS.DB“
- Conistency of DEPS.DB on several build hosts is realized with version control. Supported version control systems are git, mercurial, darcs, subversion and CVS.

How SUMO works – DEPS.DB

Example of DEPS.DB query

```
$ sumo db show AGILENT-SUPPORT:R1-12
{
    "AGILENT-SUPPORT": {
        "R1-12": {
            "aliases": {
                "BASE": "EPICS_BASE"
            },
            "dependencies": [
                "BASE",
                "BESSYRULES"
            ],
            "source": {
                "darcs": {
                    "tag": "R1-12",
                    "url": "http://repo.acc.bessy.de/darcs/epics/support/agilent-support"
                }
            }
        }
    }
}
```

How SUMO works – BUILDS.DB

Example of BUILDS:DB query:

```
$ sumo build show IDCP-011
{
    "IDCP-011": {
        "linked": {
            "RTEMS-BUILD-TOOL": "IDCP-006",
            "TOOLS_HGEN": "IDCP-001"
        },
        "modules": {
            "ALARM": "R4-1",
            ...
            "BASE": "R3-14-12-7-bessy5",
            "BESSYRULES": "R2-21",
            ...
            "RTEMS-BUILD-TOOL": "4.9-1.23-debian-7",
            "SEQ": "R2-2-3",
            ...
            "TOOLS_HGEN": "1.7",
            ...
        },
        "state": "testing"
    }
}
```

How to use SUMO: Help

- SUMO has a single command line program, „sumo“
- Command line help:
  - sumo -h : short general help
  - sumo help : show available commands
  - sumo help COMMAND [SUBCOMMAD] : show help for command
  - sumo -h OPTION : show help for option

How to use SUMO: Use a build

Example:

```
$ sumo build use
using build IDCP-011
```

- Creates file „configure/RELEASE“
- May fail if no build exists that matches configure/MODULES

How to use SUMO: Create a build

Example:

```
$ sumo build new --makeflags "-sj" --progress
creating build 'local-IDCP-001'
checking out IDCP_CAN:1.3
checking out MCAN:R2-8-18
creating makefile '/srv/projects/ctl/pfeiffer/project-stretch/ID/idcp/sandbox/build/Makefile-local-IDCP-001'
calling make -sj config
calling make -sj all
 ... some compiler warnings and messages ...
build 'local-IDCP-001' created
```

- Creates a new build
- You must still run „sumo build use“ later

How to install SUMO on your host

- Described at SUMO homepage:

  https://epics-sumo.sourceforge.io/sumo-install.html

  - Install as *.deb or *.rpm package
  - Install with „pip“
  - Install from *.tar.gz
- Described in Twiki (configuration for BESSY software repositories, sumo-bootstrap.sh script)

  http://wiki.trs.bessy.de/bin/view/Controls/Sumo

Links and references

- SUMO homepage:

  https://epics-sumo.sourceforge.io/index.html

- SUMO at BESSY Twiki Page

  http://wiki.trs.bessy.de/bin/view/Controls/SumoAtBessy

- Installing SUMO Twiki Page

  http://wiki.trs.bessy.de/bin/view/Controls/Sumo