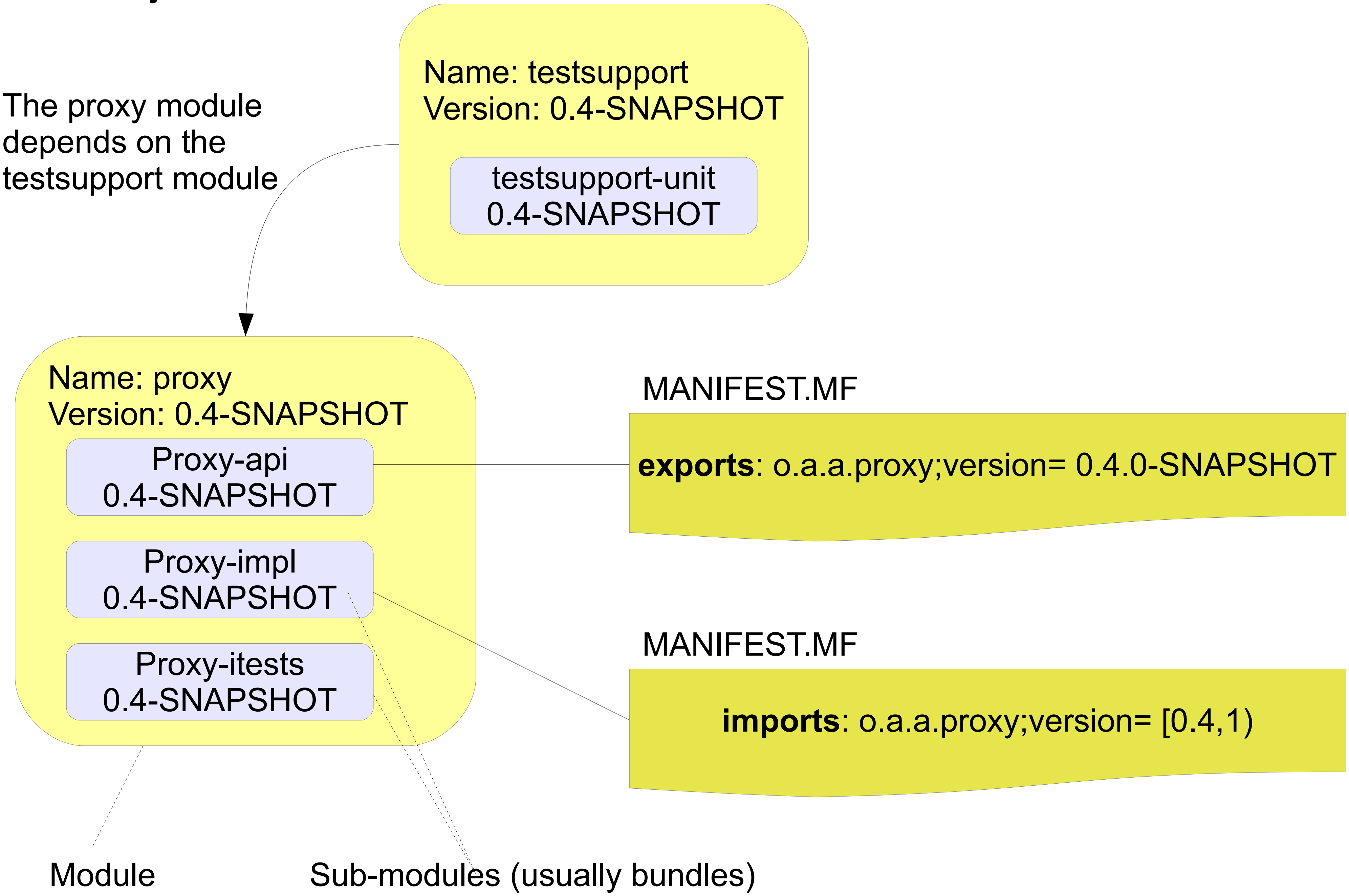
Currently in trunk



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.4-SNAPSHOT

testsupport-unit
0.4-SNAPSHOT

Name: proxy
Version: 0.4-SNAPSHOT

Proxy-api
0.4-SNAPSHOT

Proxy-impl
0.4-SNAPSHOT

Proxy-itests
0.4-SNAPSHOT

MANIFEST.MF

exports: o.a.a.proxy;version= 0.4.0-SNAPSHOT

MANIFEST.MF

imports: o.a.a.proxy;version= [0.4,1)

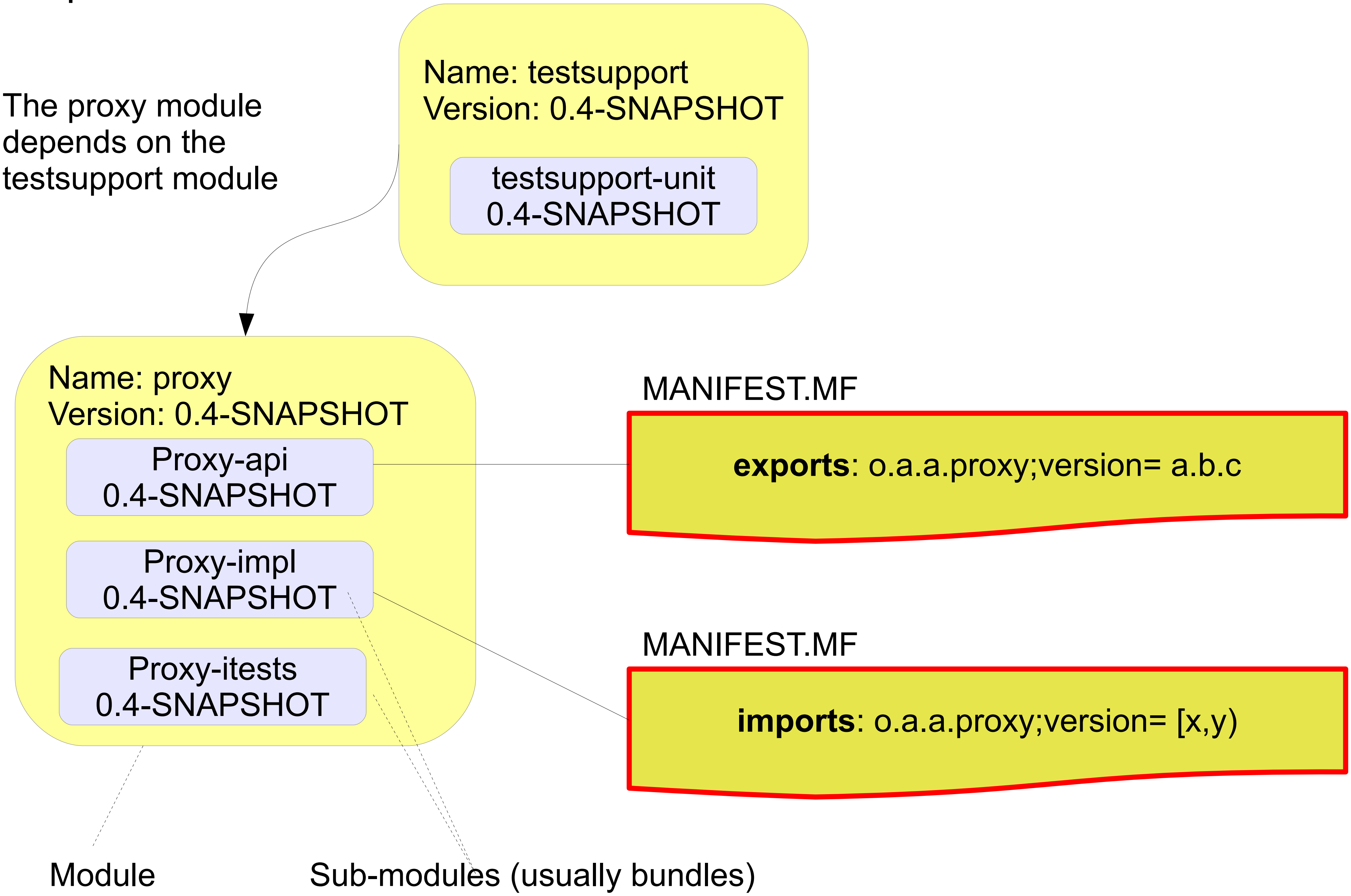
Module

Sub-modules (usually bundles)

Package vesions

- Package import and export versions are derived from the maven artifact version. This is wrong
- Exports: if we release proxy as it is now we would release an api bundle that exports o.a.a.proxy;version=0.4 even if no changes at all have been made in the proxy api.
- Imports: We import testsupport at a version derived from the version of proxy, not the version of testsupport that we build against.

Step 1



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.4-SNAPSHOT

testsupport-unit
0.4-SNAPSHOT

Name: proxy
Version: 0.4-SNAPSHOT

Proxy-api
0.4-SNAPSHOT

Proxy-impl
0.4-SNAPSHOT

Proxy-itests
0.4-SNAPSHOT

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

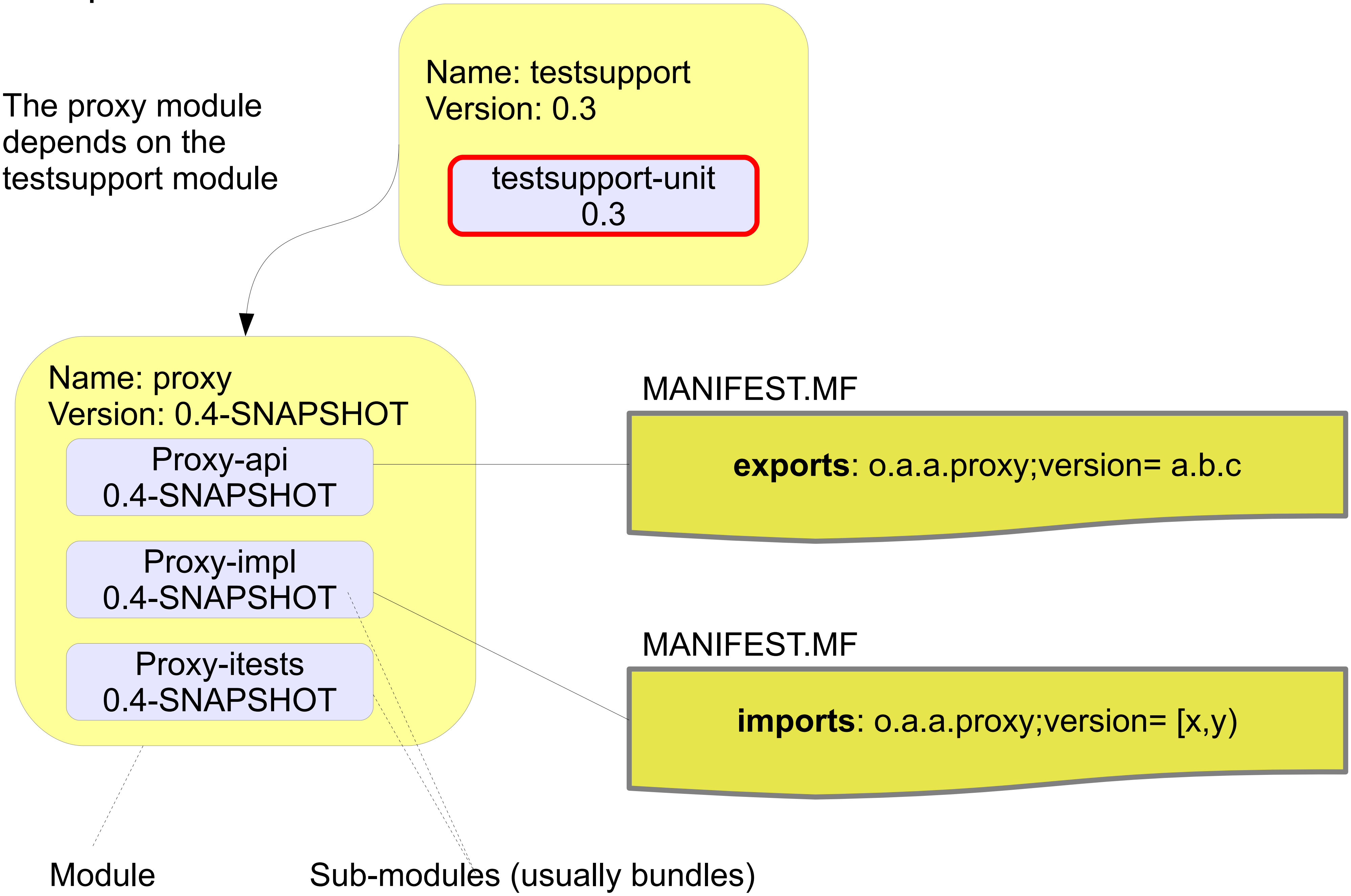
imports: o.a.a.proxy;version= [x,y)

Module

Sub-modules (usually bundles)

- As it stands proxy depends on the version of testsupport in trunk. But there is no real reason for this, if testsupport has just been released we should depend on a released version
- This has implications for development.
    - When someone makes a change in testsupport any projects which depend on it should switch to using a development version
    - Does this depend on the nature of the change? Theoretically yes, but for simplicity perhaps we should say 'any' change?

Step 2



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.3

testsupport-unit
0.3

Name: proxy
Version: 0.4-SNAPSHOT

Proxy-api
0.4-SNAPSHOT

Proxy-impl
0.4-SNAPSHOT

Proxy-itests
0.4-SNAPSHOT

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

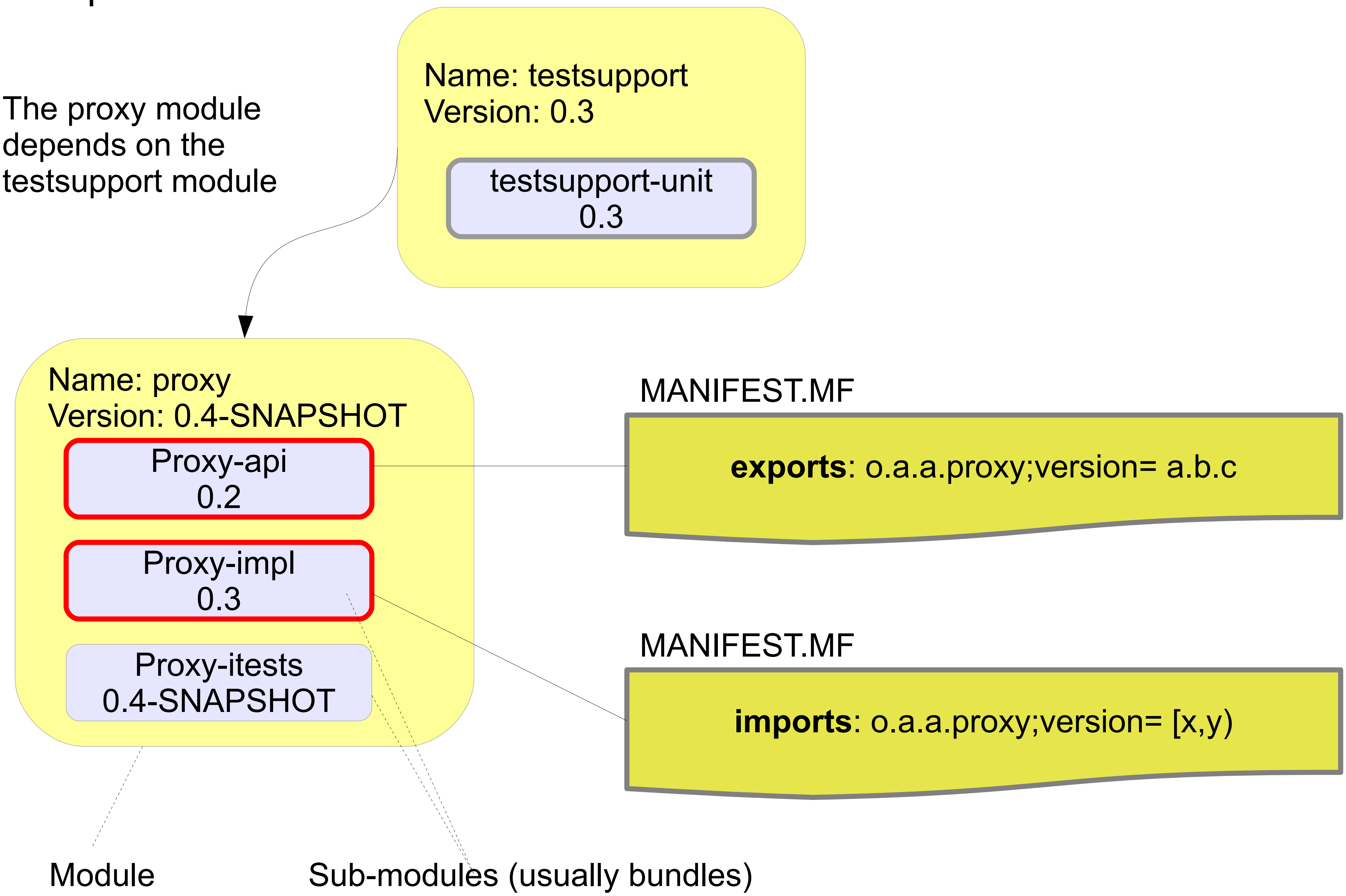
imports: o.a.a.proxy;version= [x,y)

Module

Sub-modules (usually bundles)

- The submodules of proxy all have the same version number
- There is nothing that dictates that they have to
- So, lets say we have a stable api, a stable implementation but we are working on itests

Step 4



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.3

testsupport-unit
0.3

Name: proxy
Version: 0.4-SNAPSHOT

Proxy-api
0.2

Proxy-impl
0.3

Proxy-itests
0.4-SNAPSHOT

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

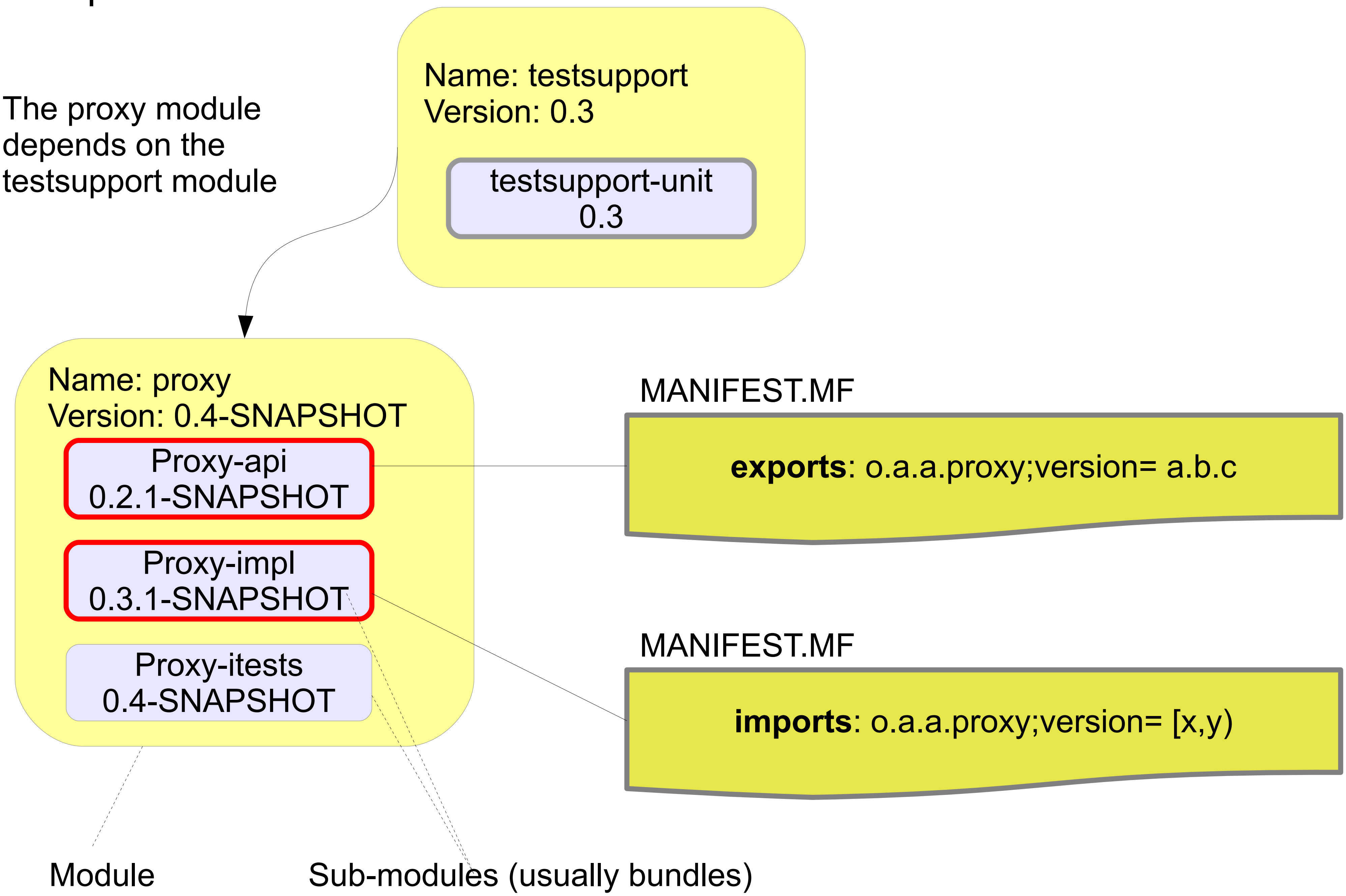
imports: o.a.a.proxy;version= [x,y)

Module

Sub-modules (usually bundles)

- If we add some itests and want to release now we'll release proxy at 0.4 (or whatever we pick) it will contain the sources for 0.2 proxy-api and 0.3-impl which are unchanged from the previous release but the only new bundle will be the itests one at version 0.4 (or whatever)
- But if during development of itests somone finds a bug in proxy-api and a bug in proxy-impl....then we would have this in trunk:

Step 5



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.3

testsupport-unit
0.3

Name: proxy
Version: 0.4-SNAPSHOT

Proxy-api
0.2.1-SNAPSHOT

Proxy-impl
0.3.1-SNAPSHOT

Proxy-itests
0.4-SNAPSHOT

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

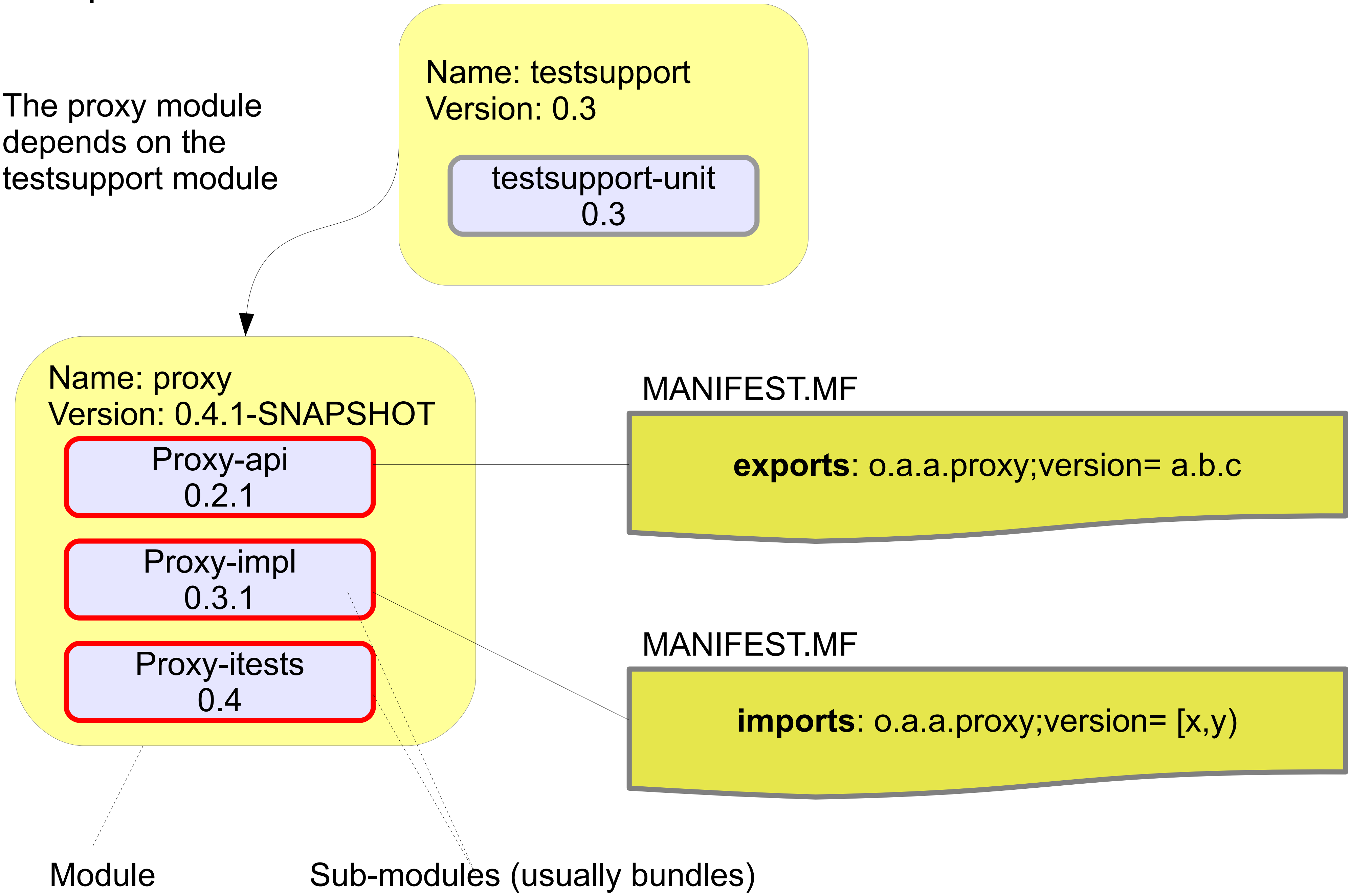
imports: o.a.a.proxy;version= [x,y)

Module

Sub-modules (usually bundles)

- The we decide to release the proxy bundle at 0.4.0 which will contain
  - Proxy-api 0.2.1
  - Proxy-impl 0.3.1
  - Proxy-itests 0.4.0
- And our new versions in trunk will be:

Step 6



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.3

testsupport-unit
0.3

Name: proxy
Version: 0.4.1-SNAPSHOT

Proxy-api
0.2.1

Proxy-impl
0.3.1

Proxy-itests
0.4

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

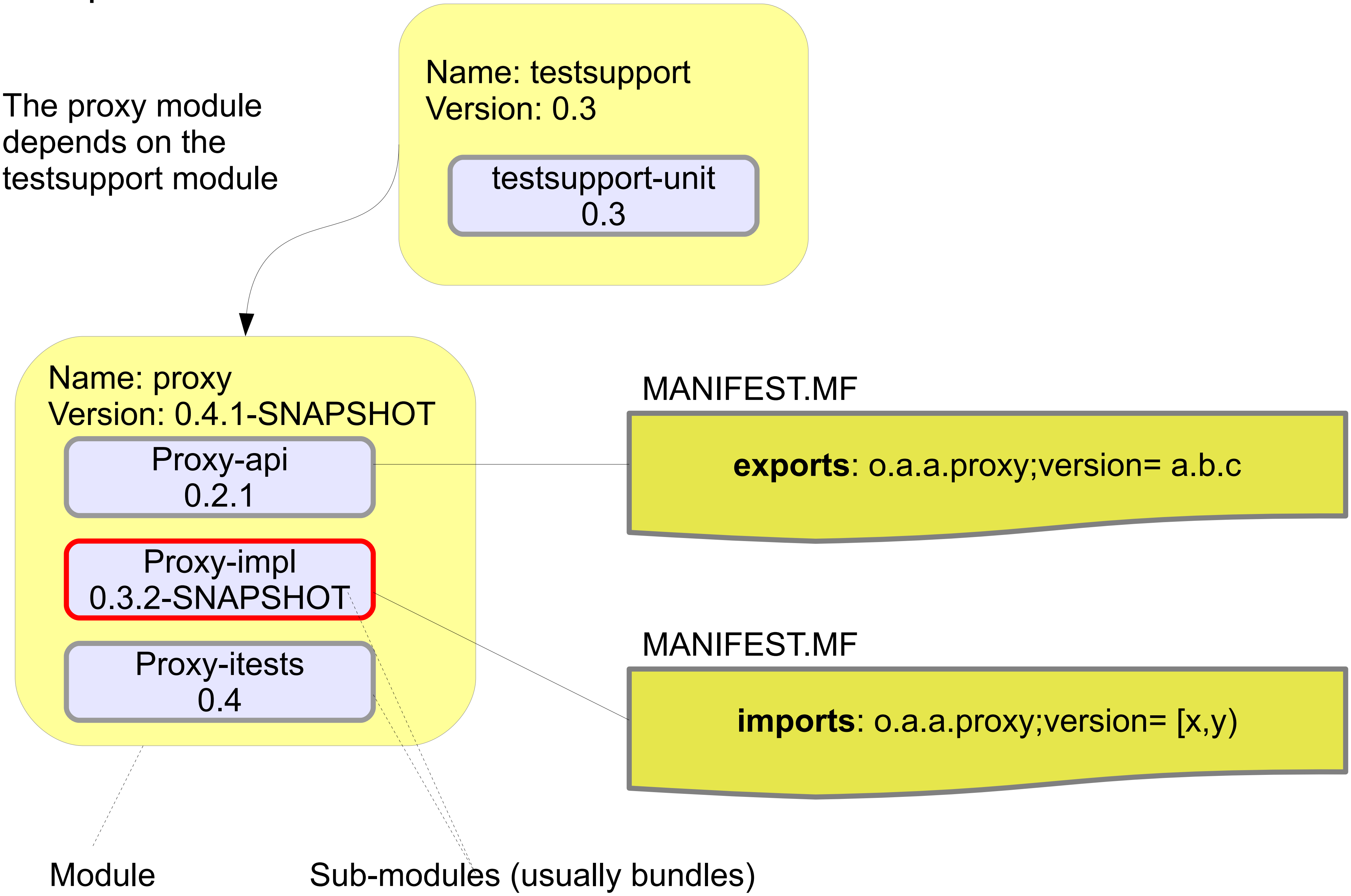
imports: o.a.a.proxy;version= [x,y)

Module

Sub-modules (usually bundles)

- Until, say, proxy-impl needs a change, then we might have

Step 7



The proxy module depends on the testsupport module

Name: testsupport
Version: 0.3

testsupport-unit
0.3

Name: proxy
Version: 0.4.1-SNAPSHOT

Proxy-api
0.2.1

Proxy-impl
0.3.2-SNAPSHOT

Proxy-itests
0.4

MANIFEST.MF

exports: o.a.a.proxy;version= a.b.c

MANIFEST.MF

imports: o.a.a.proxy;version= [x,y)

Module

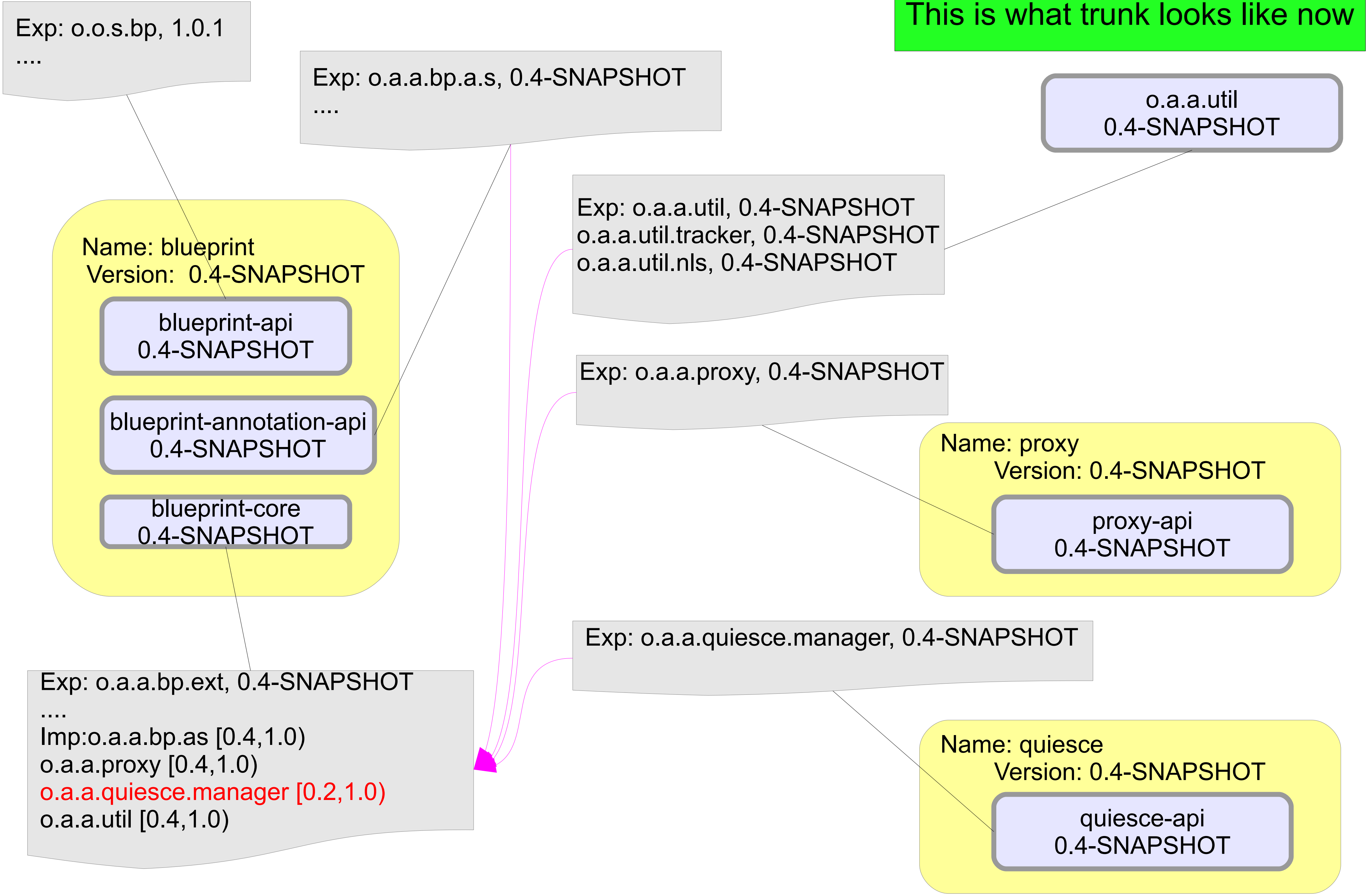
Sub-modules (usually bundles)

- The question is, will this be too confusing?
- If so, the answer is to insist that all a modules submodules stay at the same version as each other, even if some of them have not changed.
- This looks much cleaner in trunk, but it breaks (I think) semantic versioning by bundle.

Release each bundle on its own?

- Eg blueprint – 12 bundles
- Separate vote on each?
- Separate JIRA component?
- Separate documentation and release notes?

This is what trunk looks like now



Exp: o.o.s.bp, 1.0.1
....

Exp: o.a.a.bp.a.s, 0.4-SNAPSHOT
....

o.a.a.util
0.4-SNAPSHOT

Name: blueprint
Version: 0.4-SNAPSHOT

blueprint-api
0.4-SNAPSHOT

blueprint-annotation-api
0.4-SNAPSHOT

blueprint-core
0.4-SNAPSHOT

Exp: o.a.a.util, 0.4-SNAPSHOT
o.a.a.util.tracker, 0.4-SNAPSHOT
o.a.a.util.nls, 0.4-SNAPSHOT

Exp: o.a.a.proxy, 0.4-SNAPSHOT

Name: proxy
Version: 0.4-SNAPSHOT

proxy-api
0.4-SNAPSHOT

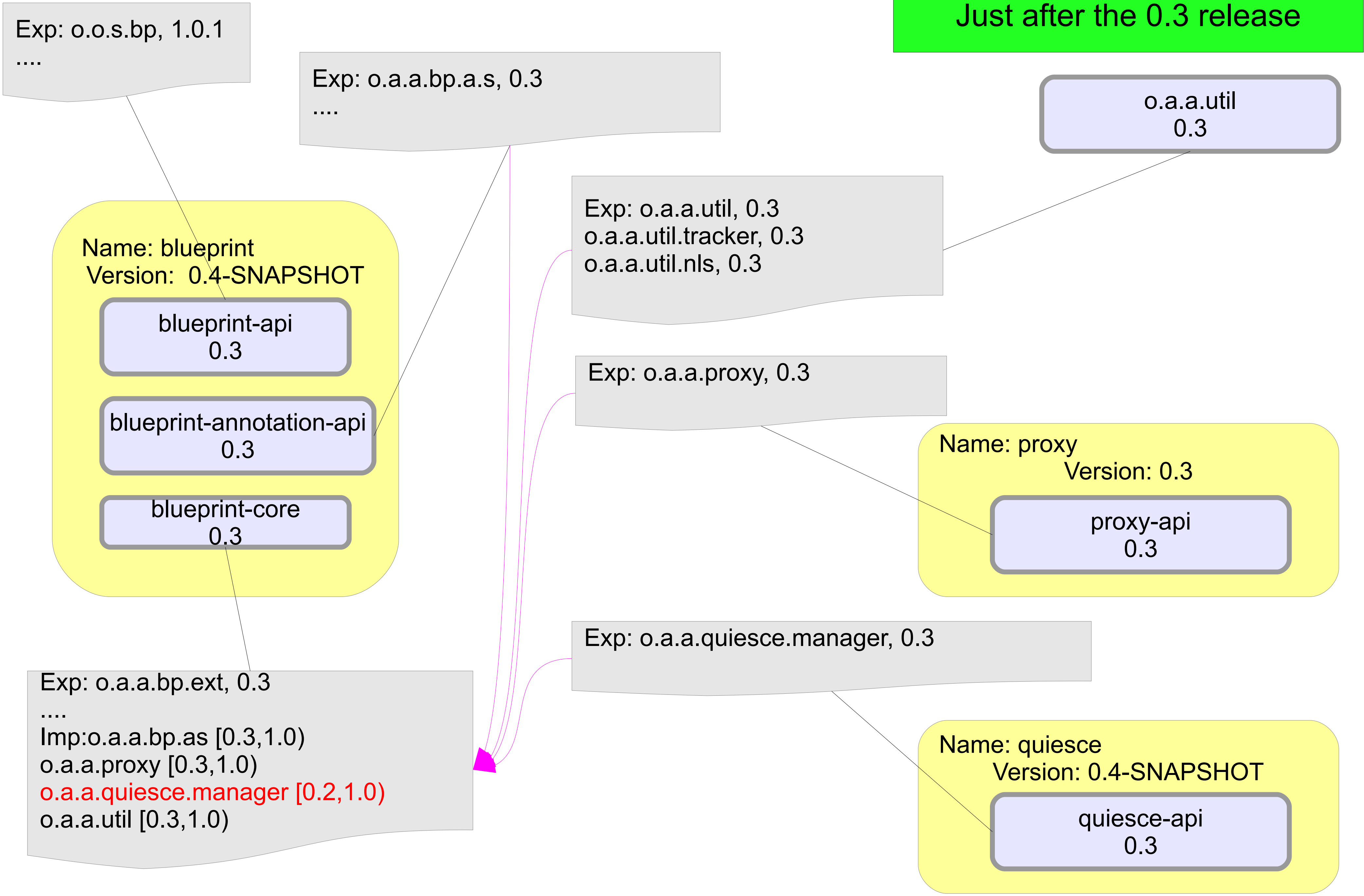
Exp: o.a.a.quiesce.manager, 0.4-SNAPSHOT

Exp: o.a.a.bp.ext, 0.4-SNAPSHOT
....
Imp:o.a.a.bp.as [0.4,1.0)
o.a.a.proxy [0.4,1.0)
o.a.a.quiesce.manager [0.2,1.0)
o.a.a.util [0.4,1.0)

Name: quiesce
Version: 0.4-SNAPSHOT

quiesce-api
0.4-SNAPSHOT

Just after the 0.3 release



Exp: o.o.s.bp, 1.0.1
....

Exp: o.a.a.bp.a.s, 0.3
....

o.a.a.util
0.3

Name: blueprint
Version: 0.4-SNAPSHOT

blueprint-api
0.3

blueprint-annotation-api
0.3

blueprint-core
0.3

Exp: o.a.a.util, 0.3
o.a.a.util.tracker, 0.3
o.a.a.util.nls, 0.3

Exp: o.a.a.proxy, 0.3

Name: proxy
Version: 0.3

proxy-api
0.3

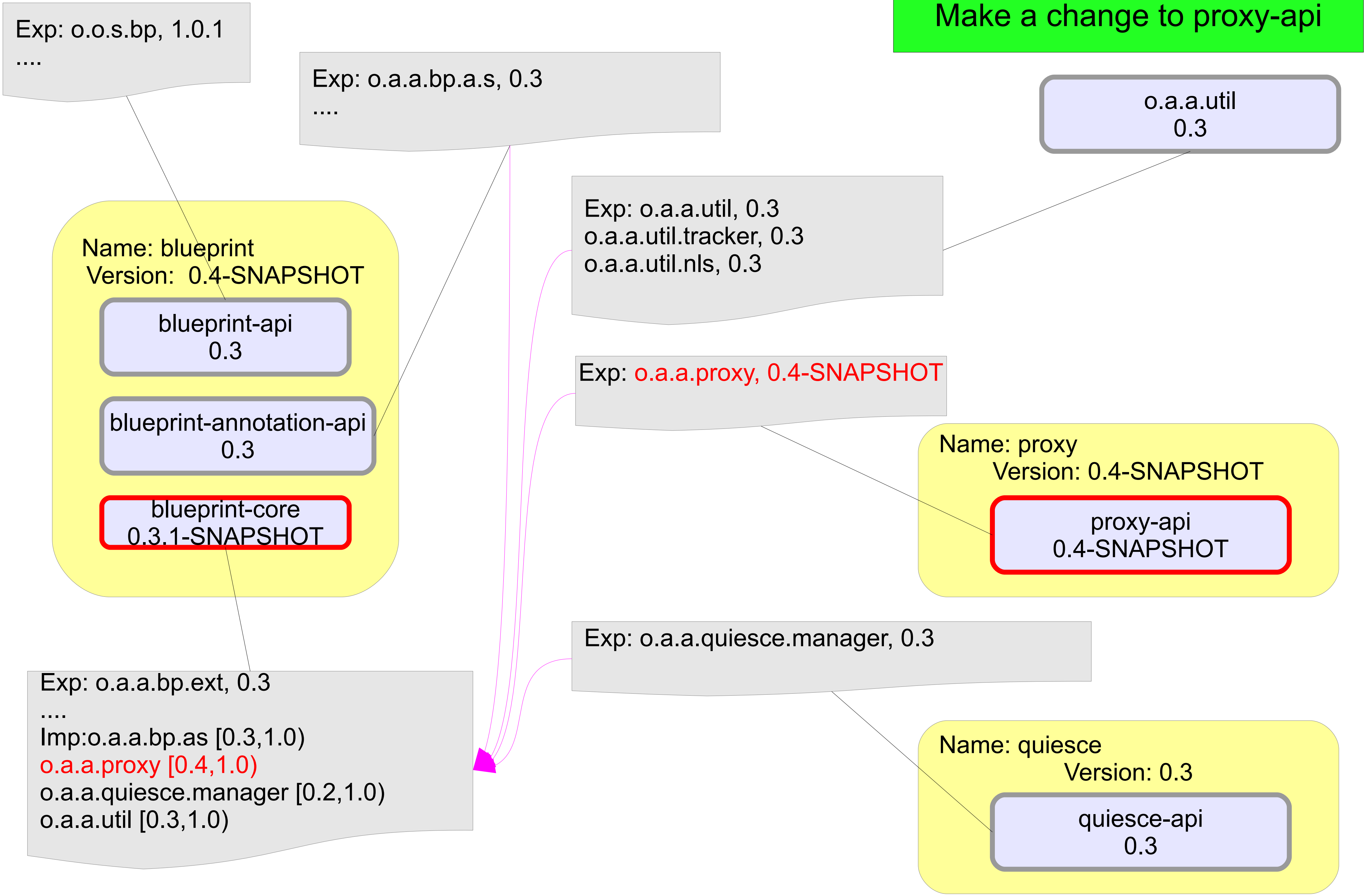
Exp: o.a.a.quiesce.manager, 0.3

Exp: o.a.a.bp.ext, 0.3
....
Imp:o.a.a.bp.as [0.3,1.0)
o.a.a.proxy [0.3,1.0)
o.a.a.quiesce.manager [0.2,1.0)
o.a.a.util [0.3,1.0)

Name: quiesce
Version: 0.4-SNAPSHOT

quiesce-api
0.3

Make a change to proxy-api



Exp: o.o.s.bp, 1.0.1
....

Exp: o.a.a.bp.a.s, 0.3
....

o.a.a.util
0.3

Exp: o.a.a.util, 0.3
o.a.a.util.tracker, 0.3
o.a.a.util.nls, 0.3

Name: blueprint
Version: 0.4-SNAPSHOT

blueprint-api
0.3

blueprint-annotation-api
0.3

blueprint-core
0.3.1-SNAPSHOT

Exp: o.a.a.proxy, 0.4-SNAPSHOT

Name: proxy
Version: 0.4-SNAPSHOT

proxy-api
0.4-SNAPSHOT

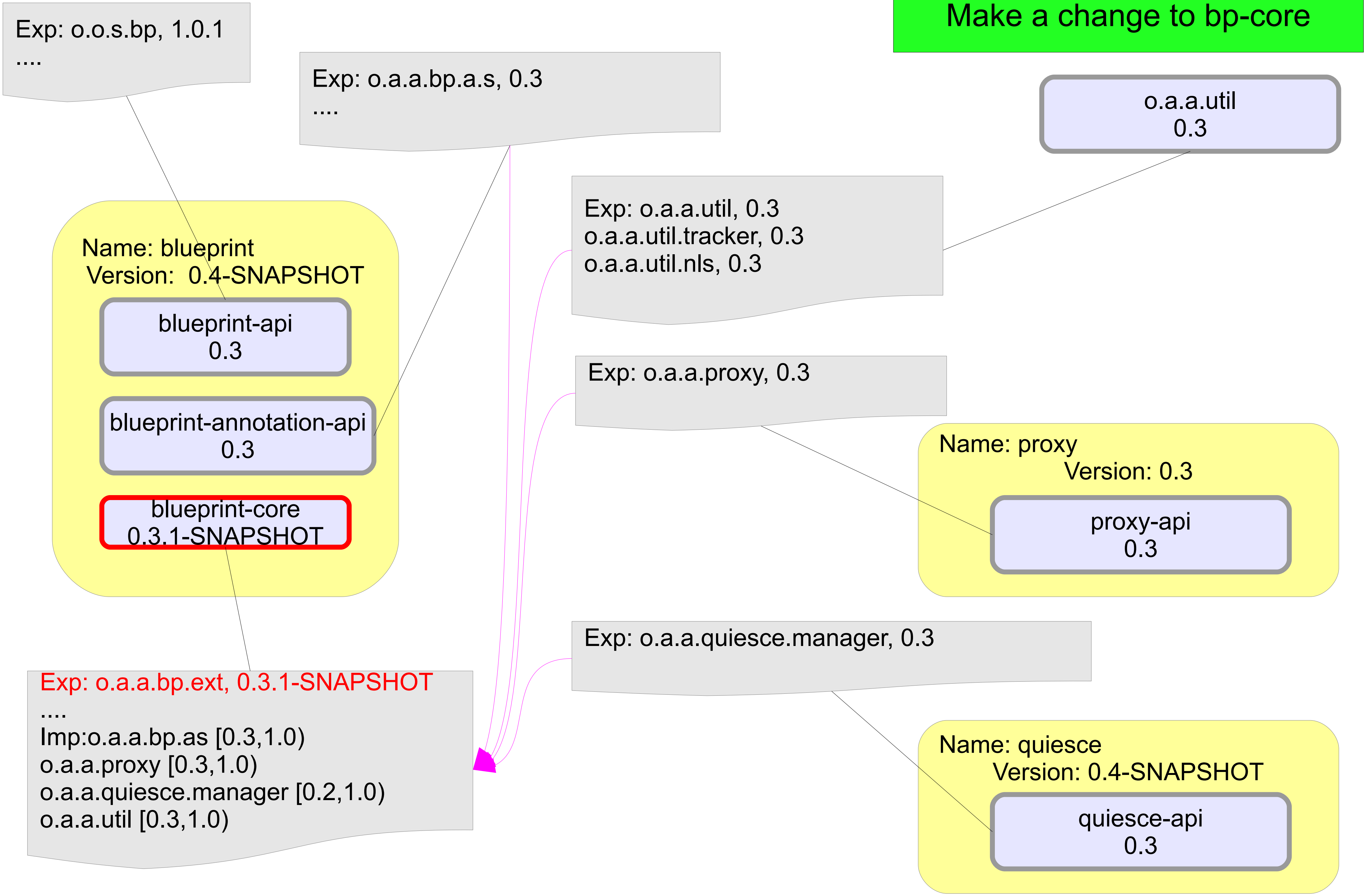
Exp: o.a.a.quiesce.manager, 0.3

Exp: o.a.a.bp.ext, 0.3
....
Imp:o.a.a.bp.as [0.3,1.0)
o.a.a.proxy [0.4,1.0)
o.a.a.quiesce.manager [0.2,1.0)
o.a.a.util [0.3,1.0)

Name: quiesce
Version: 0.3

quiesce-api
0.3

Make a change to bp-core



Exp: o.o.s.bp, 1.0.1
....

Exp: o.a.a.bp.a.s, 0.3
....

o.a.a.util
0.3

Name: blueprint
Version: 0.4-SNAPSHOT

blueprint-api
0.3

blueprint-annotation-api
0.3

blueprint-core
0.3.1-SNAPSHOT

Exp: o.a.a.util, 0.3
o.a.a.util.tracker, 0.3
o.a.a.util.nls, 0.3

Exp: o.a.a.proxy, 0.3

Name: proxy
Version: 0.3

proxy-api
0.3

Exp: o.a.a.quiesce.manager, 0.3

Name: quiesce
Version: 0.4-SNAPSHOT

quiesce-api
0.3

Exp: o.a.a.bp.ext, 0.3.1-SNAPSHOT
....
Imp:o.a.a.bp.as [0.3,1.0)
o.a.a.proxy [0.3,1.0)
o.a.a.quiesce.manager [0.2,1.0)
o.a.a.util [0.3,1.0)