

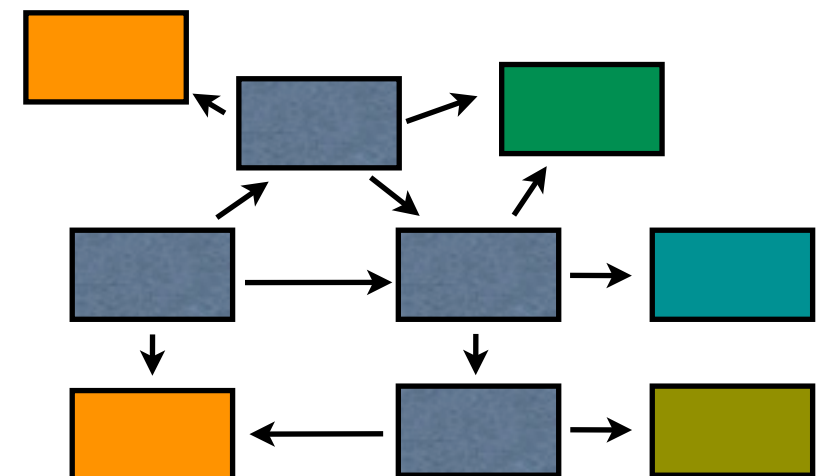
Abhängigkeiten und Git

Continuous Lifecycle 2013, 11.11.2013

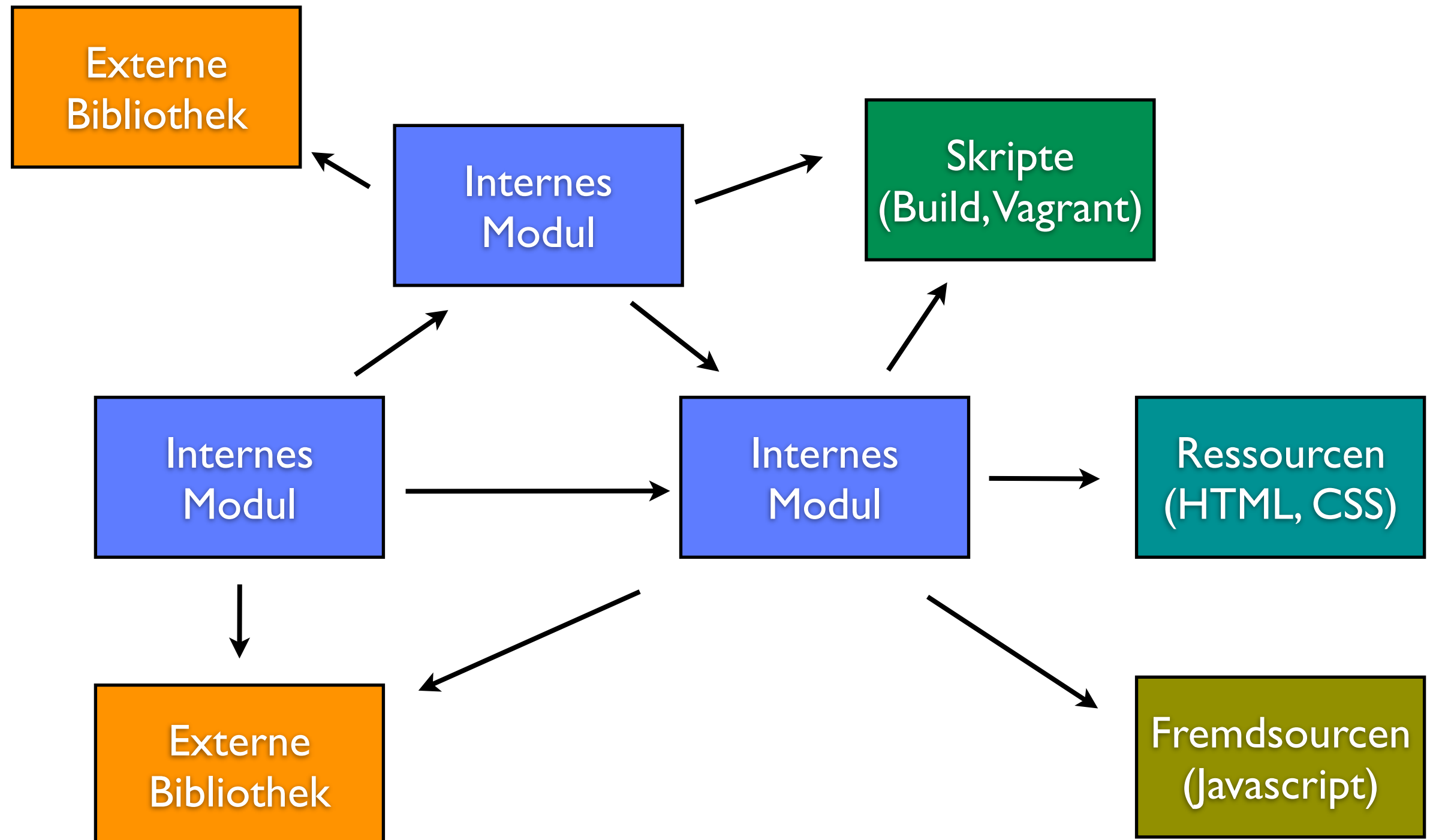
René Preißel (@RenePreissel)
Bjørn Stachmann (@old_stachi)

Inhalt

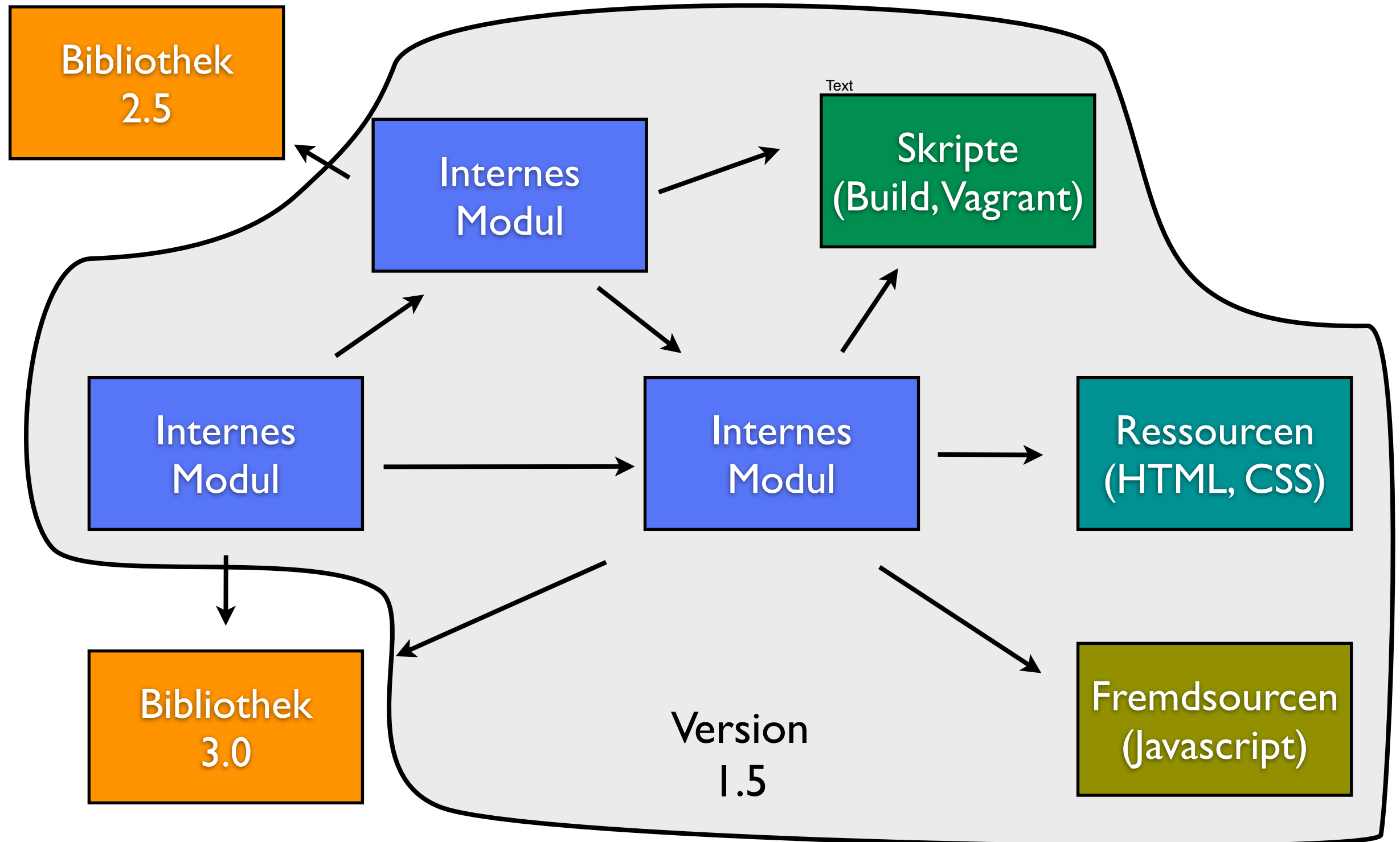
1. Über welche Abhängigkeiten reden wir?
2. Wie versionieren?
 - a. Ein großes Repository
 - b. Viele kleine Repositories
 - c. Irgendwas dazwischen
3. Beispiele aus der Praxis



Typisches Projekt



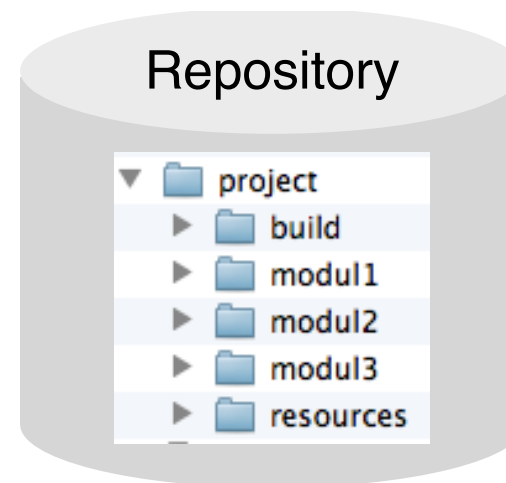
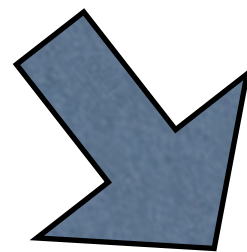
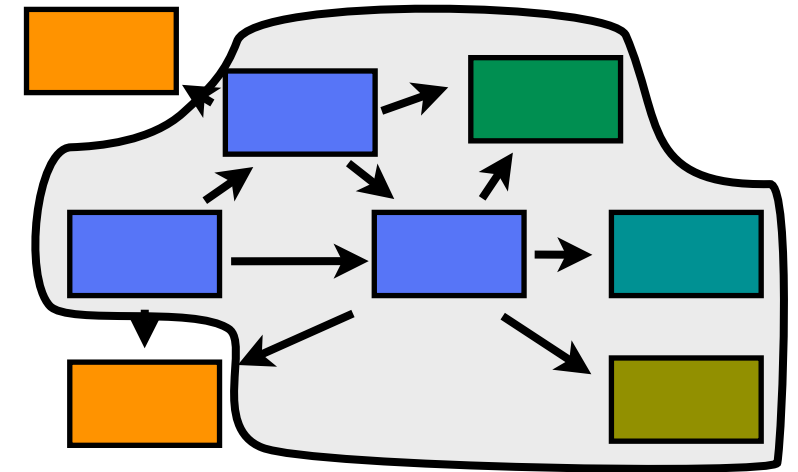
Eine Releaseeinheit





Eine Release-Einheit für eigene Entwicklung

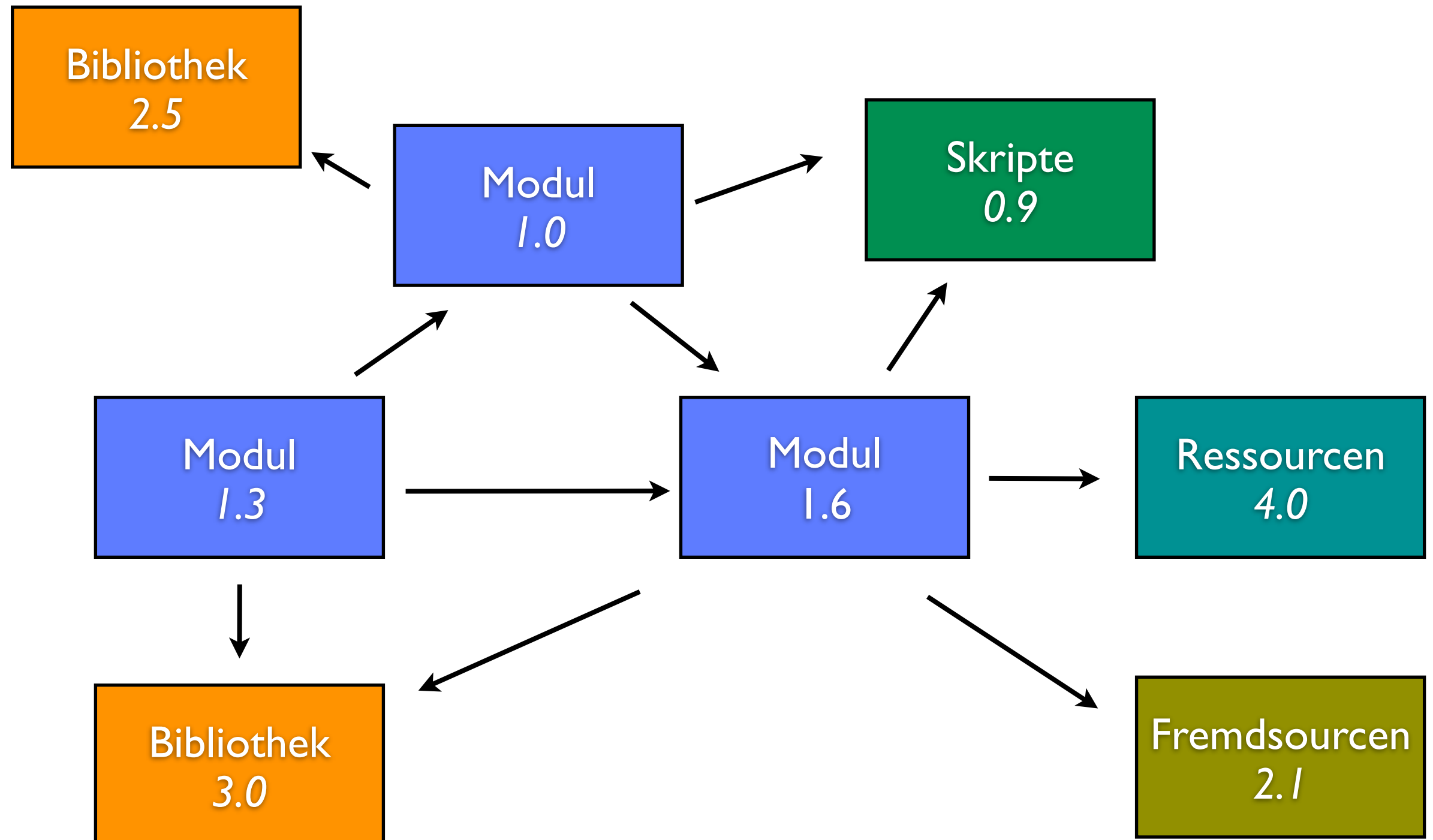
- Eine Version
- Ein Lebenszyklus



Ein
Repository

- Tags / Branches
für alle Module

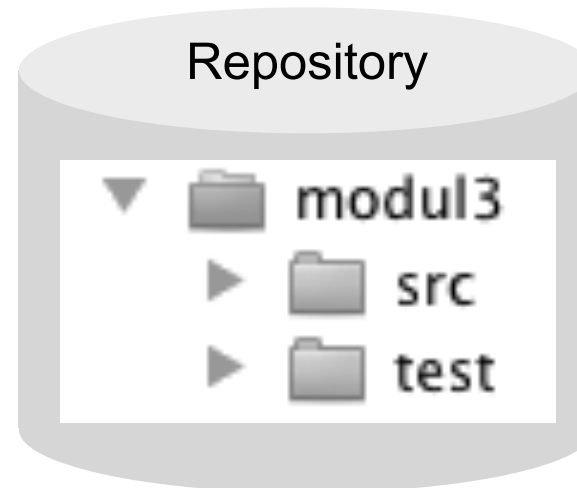
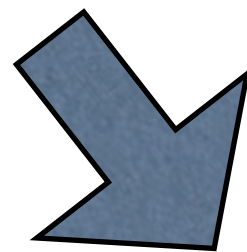
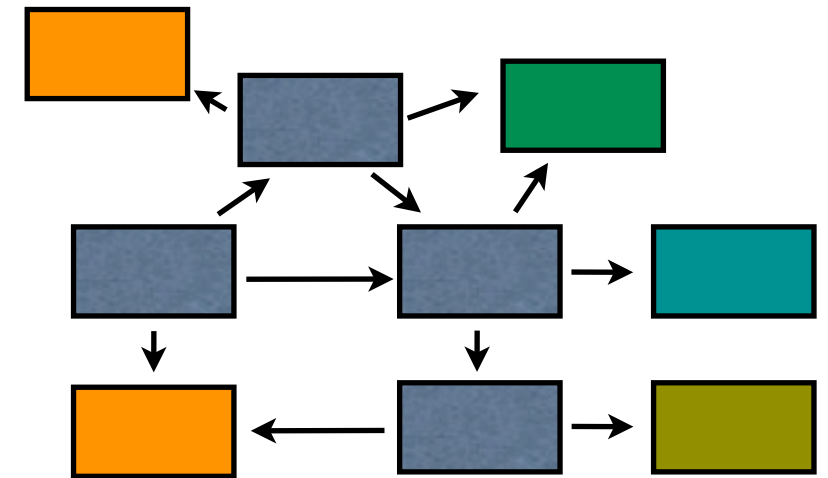
Jedes Modul eine Releaseeinheit





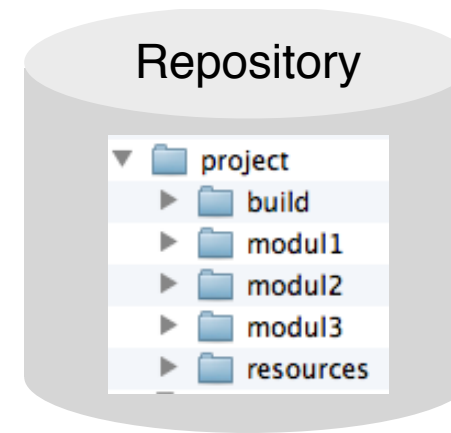
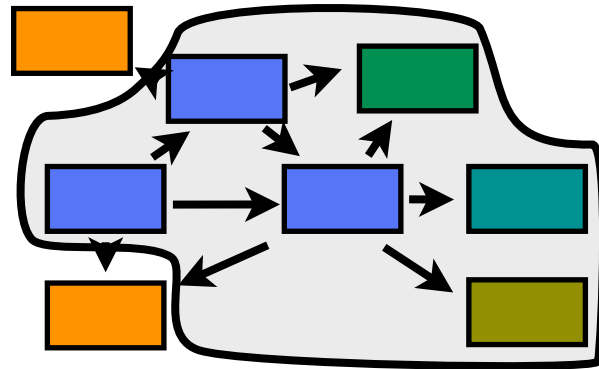
Eine Release-Einheit pro Modul

- Eigene Version
- Eigener Lebenszyklus

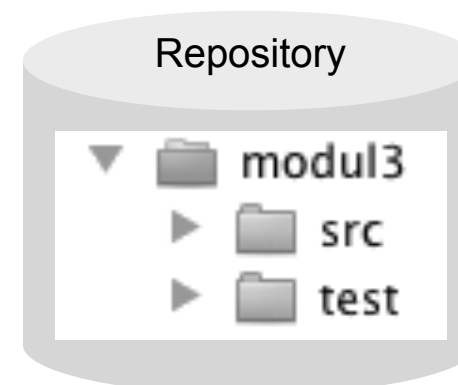
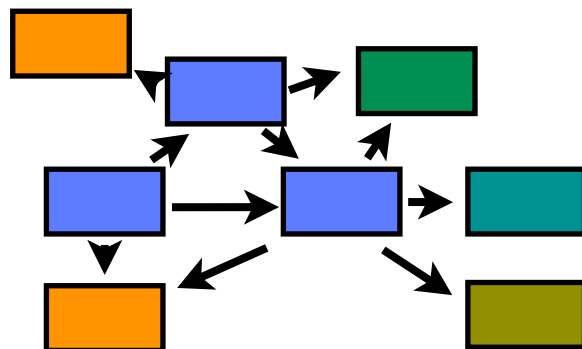


Eigenes Repository

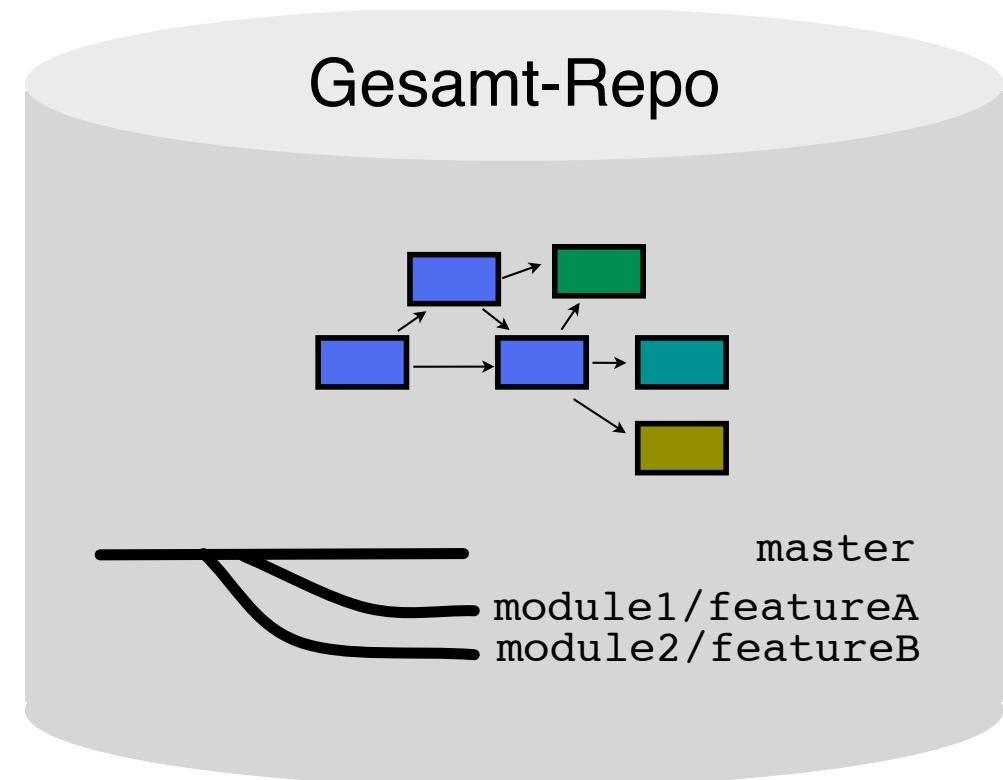
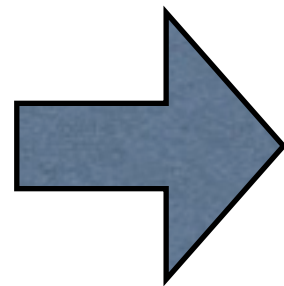
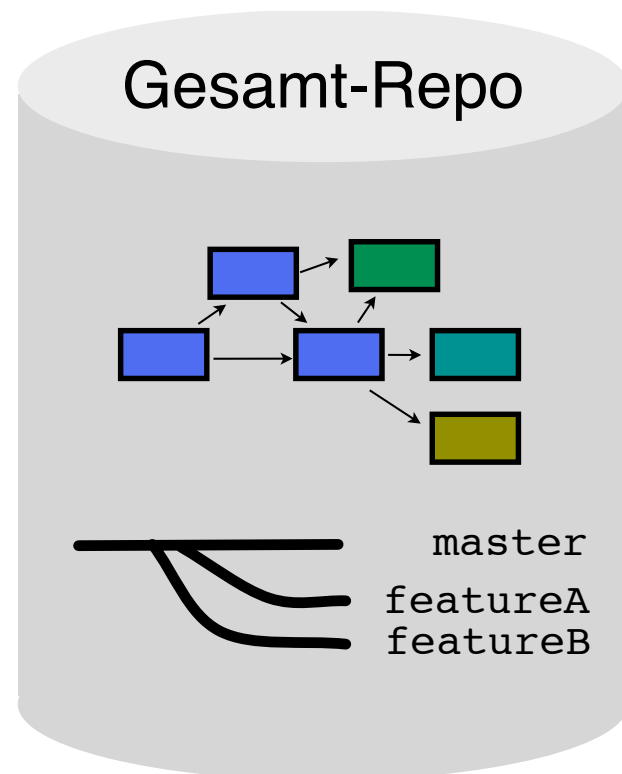
- Eigene Tags und Branches für jedes Modul

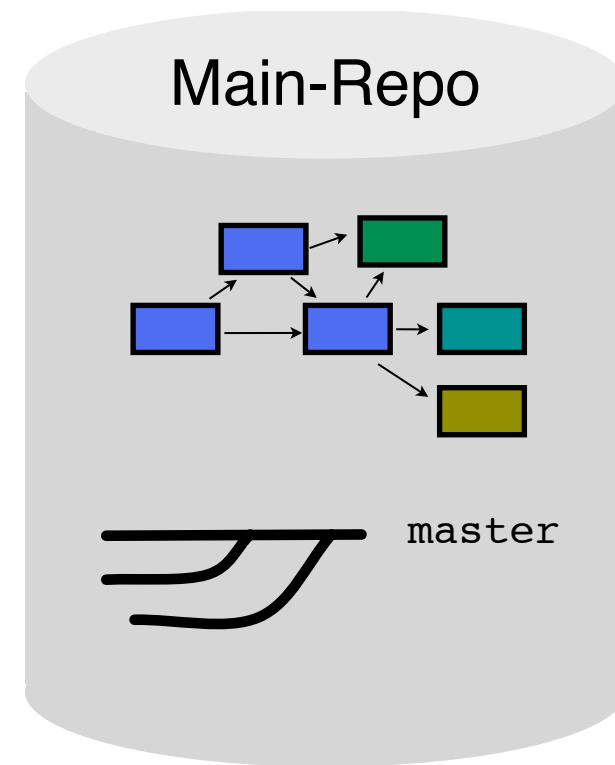
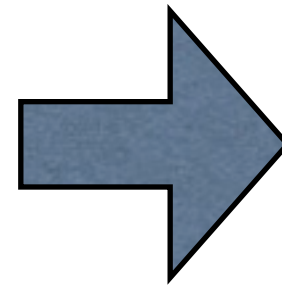
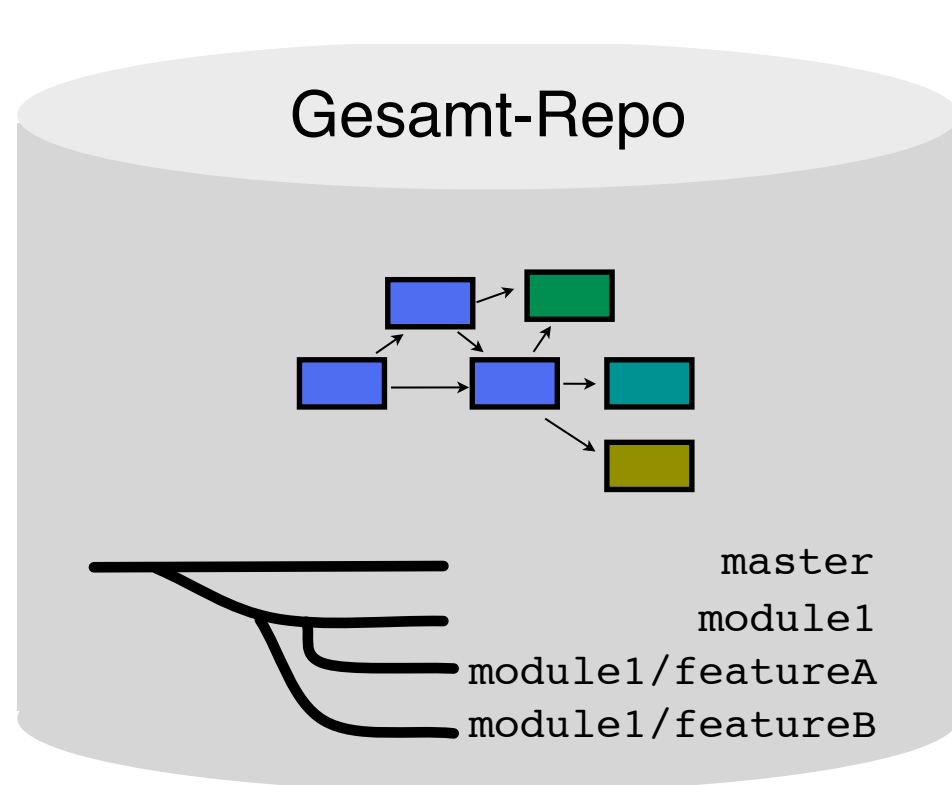


- Modul-übergreifende Features / Refactoring
- Globales Build
- Einfaches Tooling
- Teilen und Weitergeben einzelner Module
- Modul-abhängiges Branching

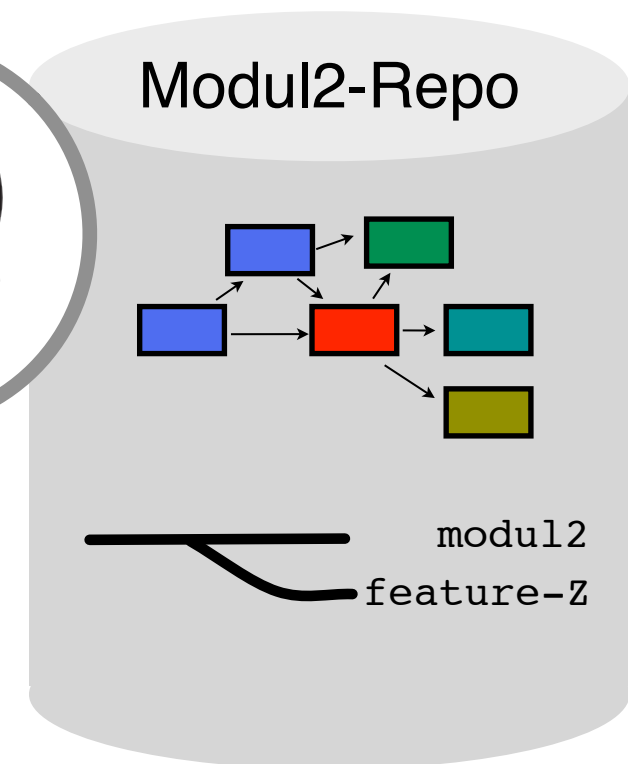
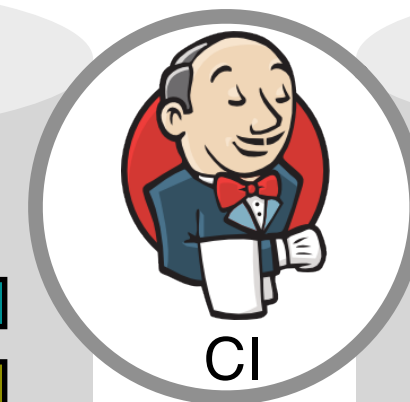
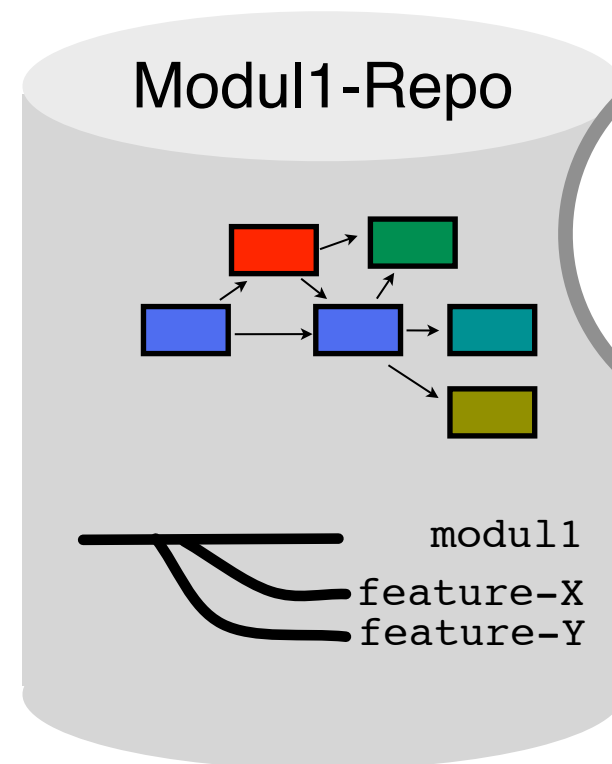


Szenarien - Ein Repo



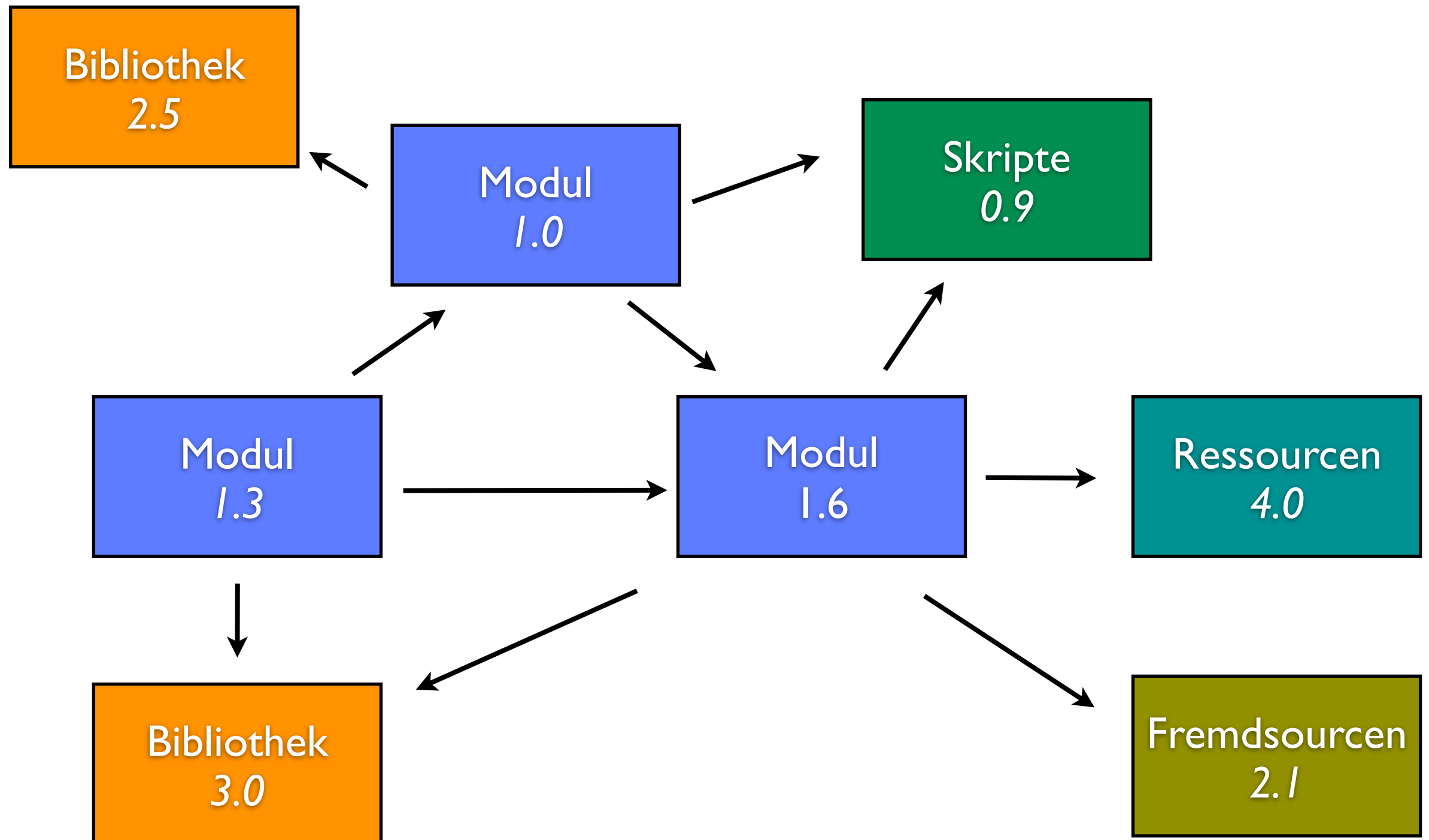


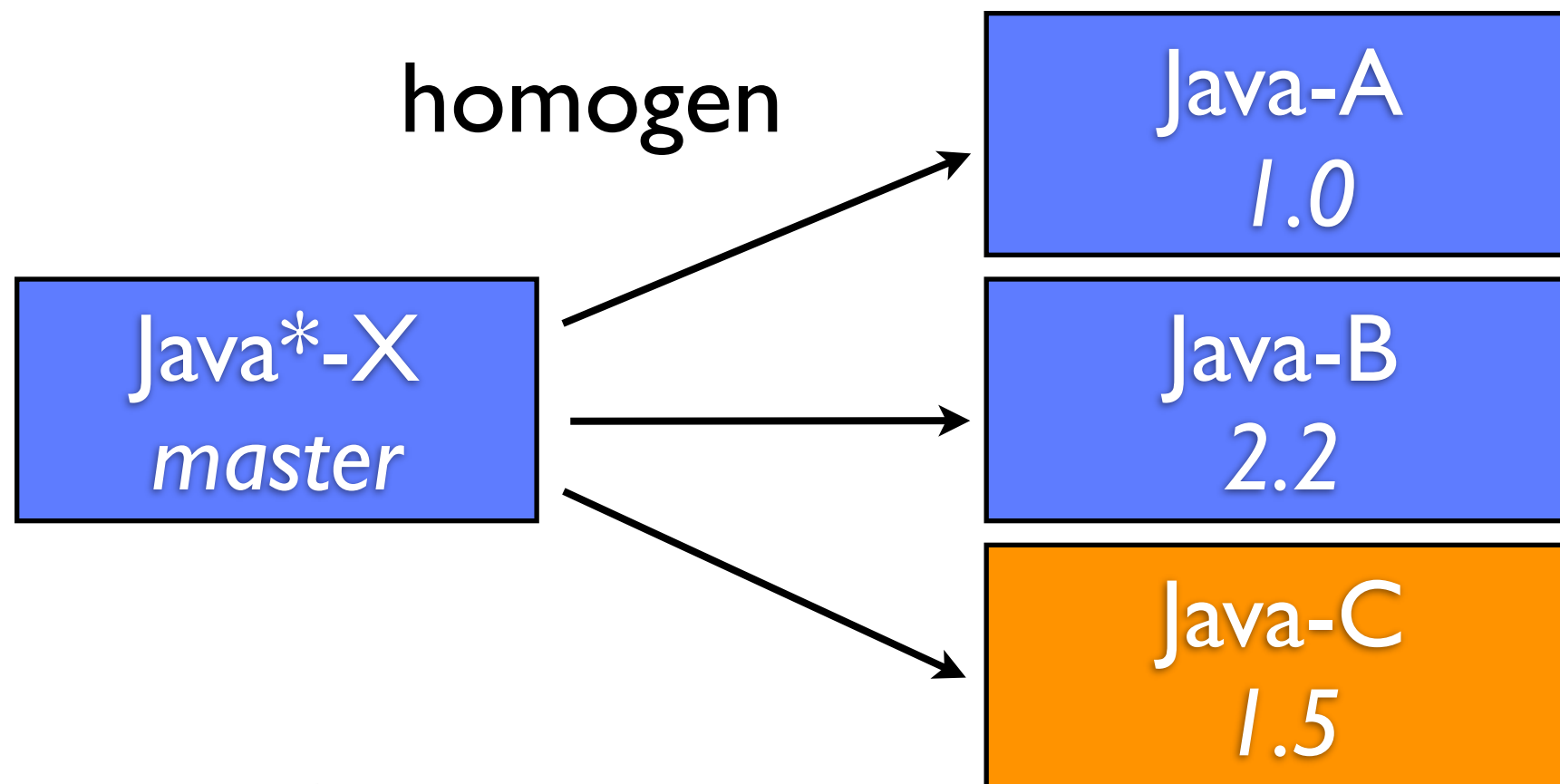
push / pull



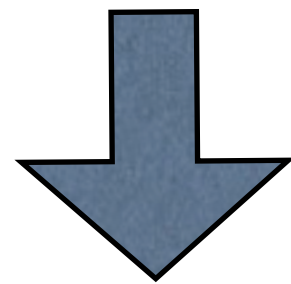
- Modul-übergreifende Features
- Refactoring zwischen Modulen

Szenarien - Repo pro Modul



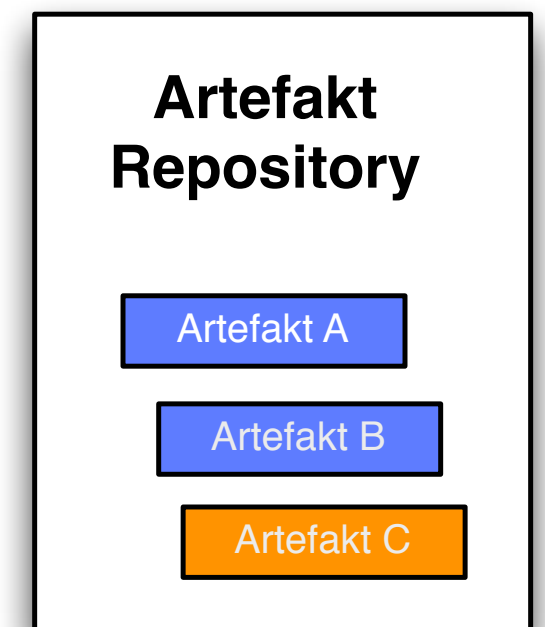
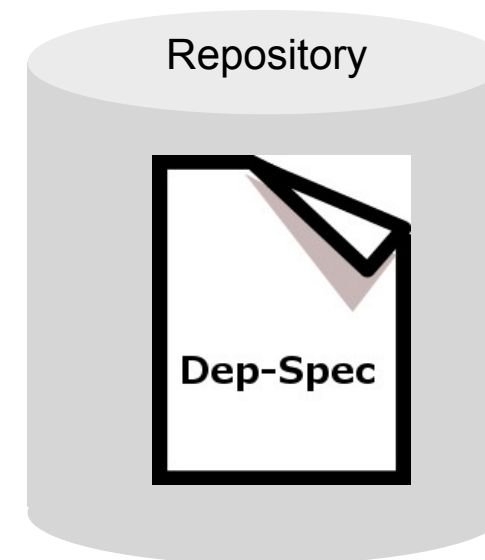
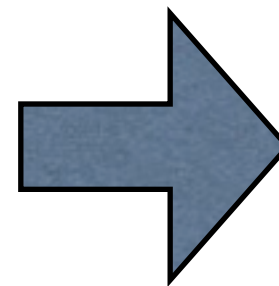


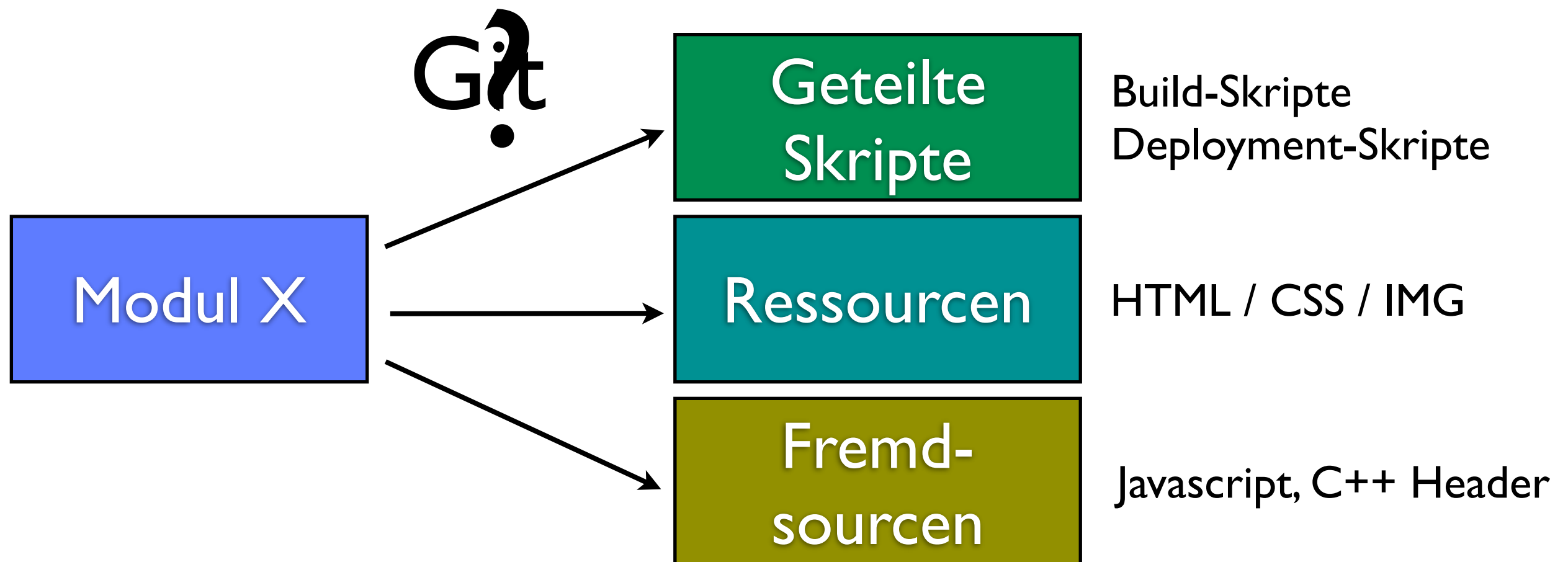
* Ersetzen mit der Technologie Ihrer Wahl



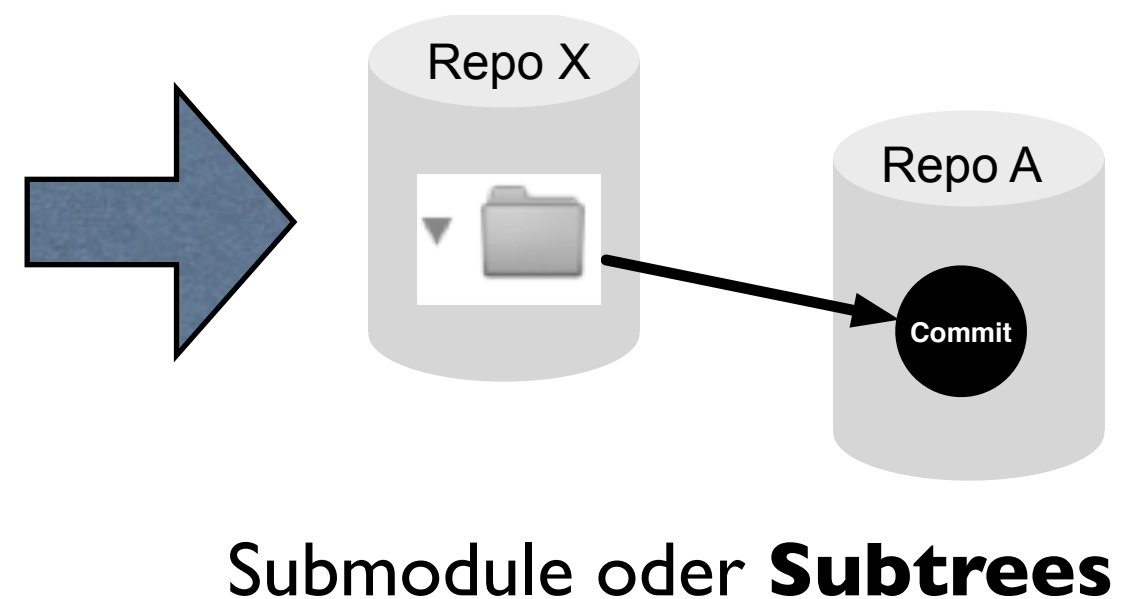
Dependency Manager

- Maven, Ivy, Gradle,
- npm, RequireJS,
- Leiningen, SBT, ...

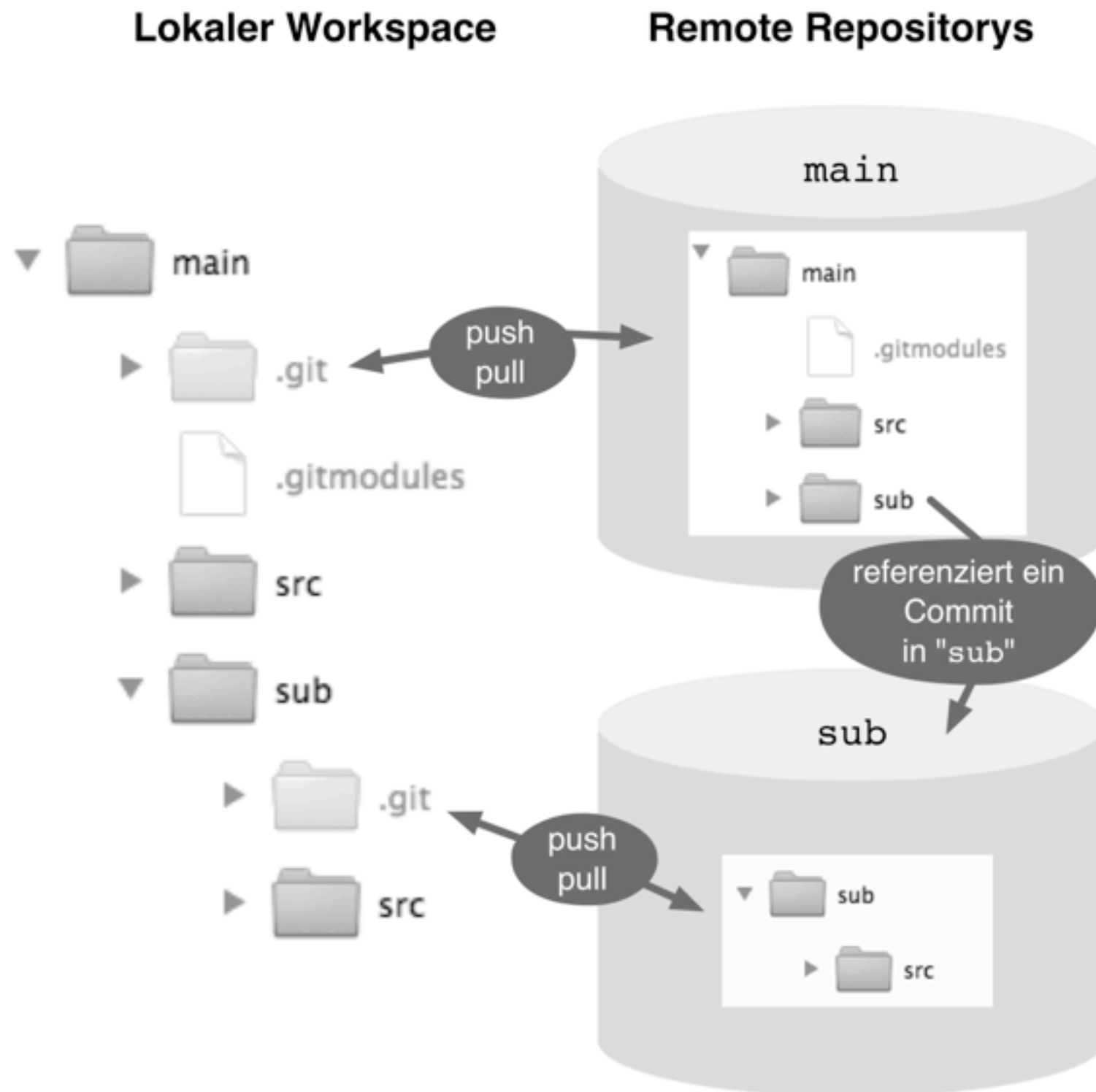




- Inhomogene Infrastruktur
- Einbindung von Sourcen und Ressourcen
- Globales Build erforderlich



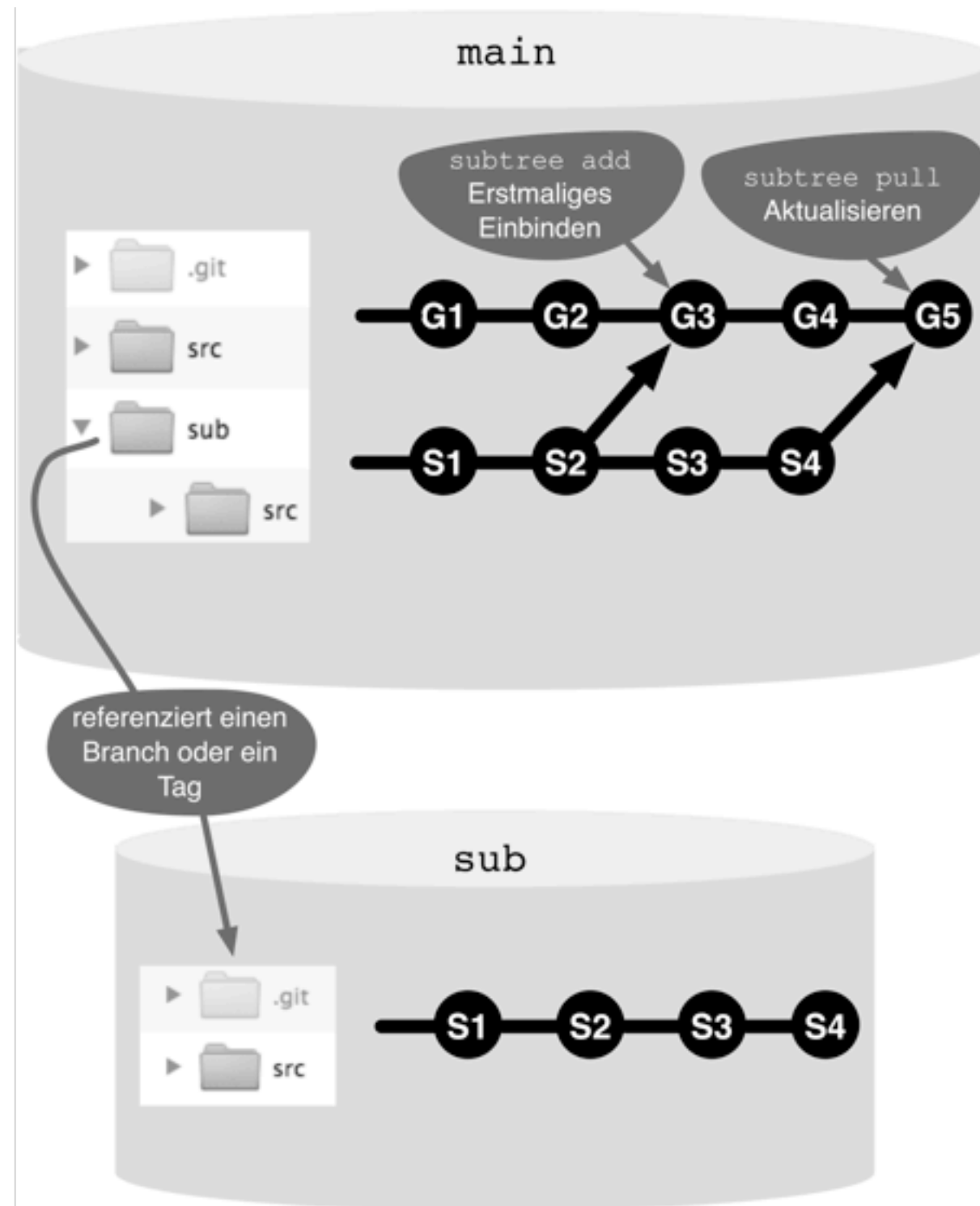
Submodule



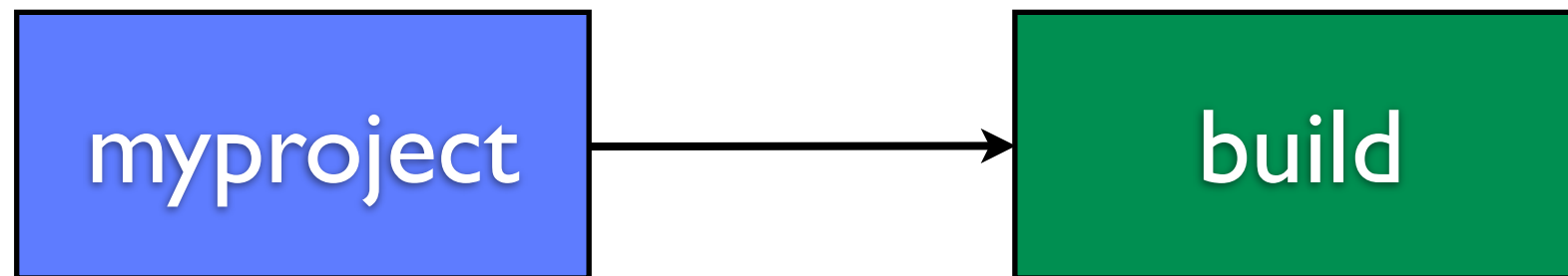
**Hohe
Komplexität**

Fragil

Subtree



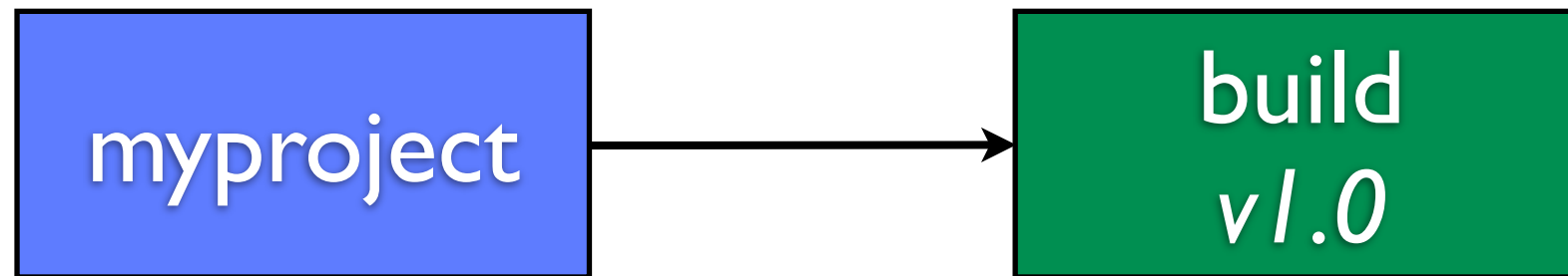
Beispiel Subtrees



Schritte:

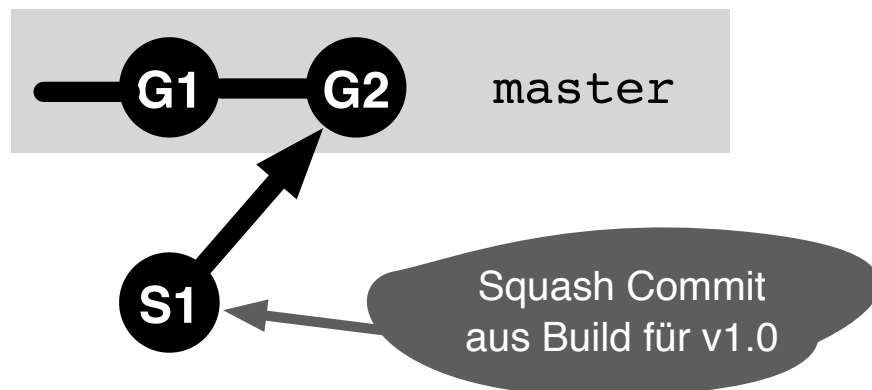
- Modul hinzufügen
- Repository mit Modulen klonen
- Neue Version eines Moduls einbinden
- Änderungen in einem Modul durchführen
- Änderungen aufsplitten und zurückübertragen

Subtree hinzufügen

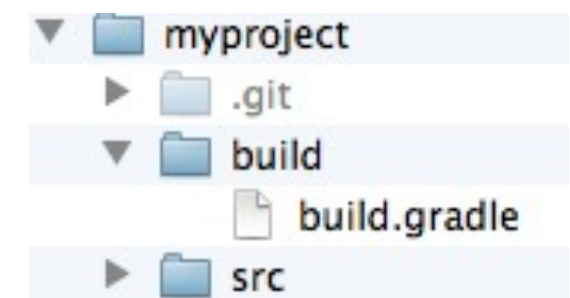
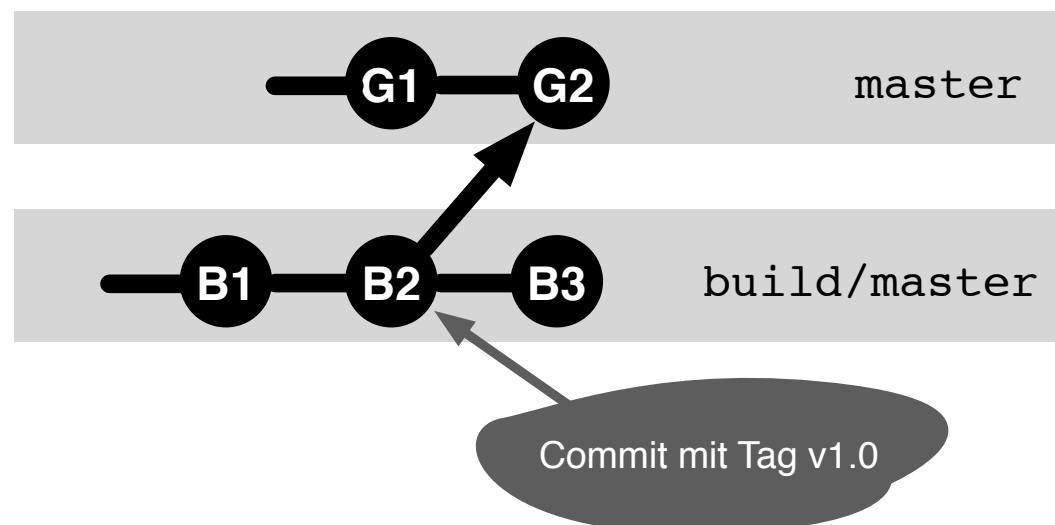


```
myproject$ git subtree add --prefix build --squash ../build.git v1.0  
myproject$ git push
```

mit --squash

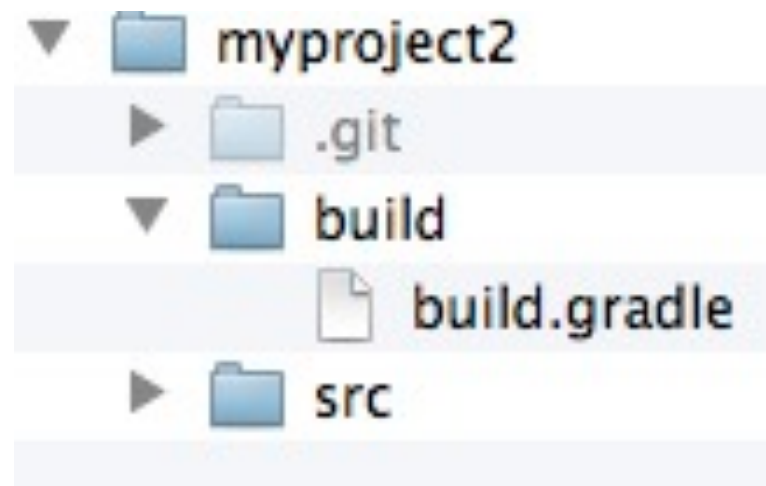


ohne --squash

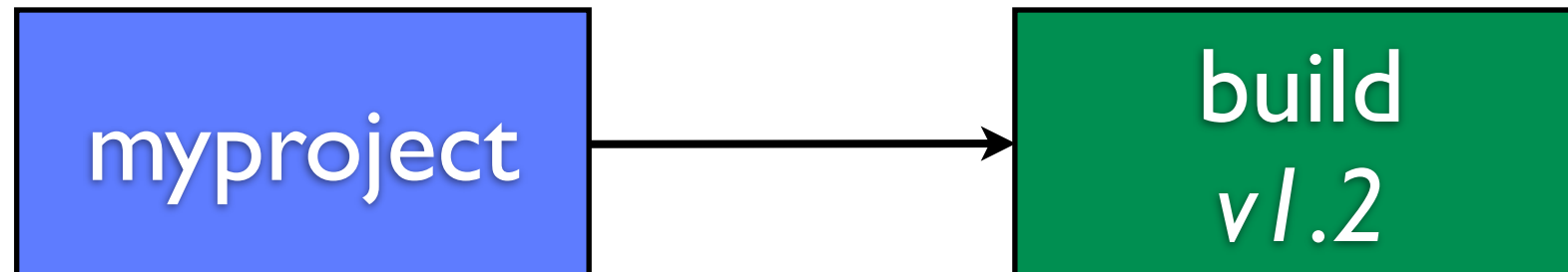


Klonen mit Subtrees

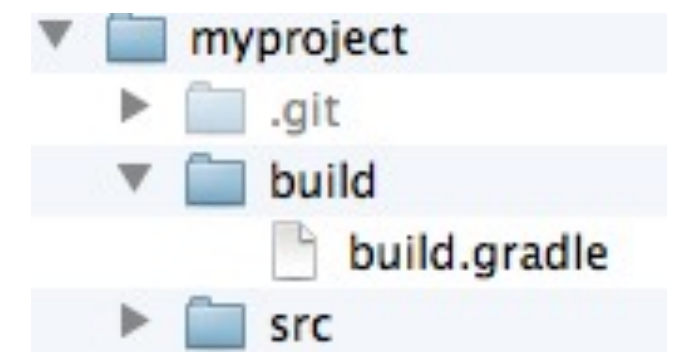
```
projects$ git clone myproject.git myproject2
```



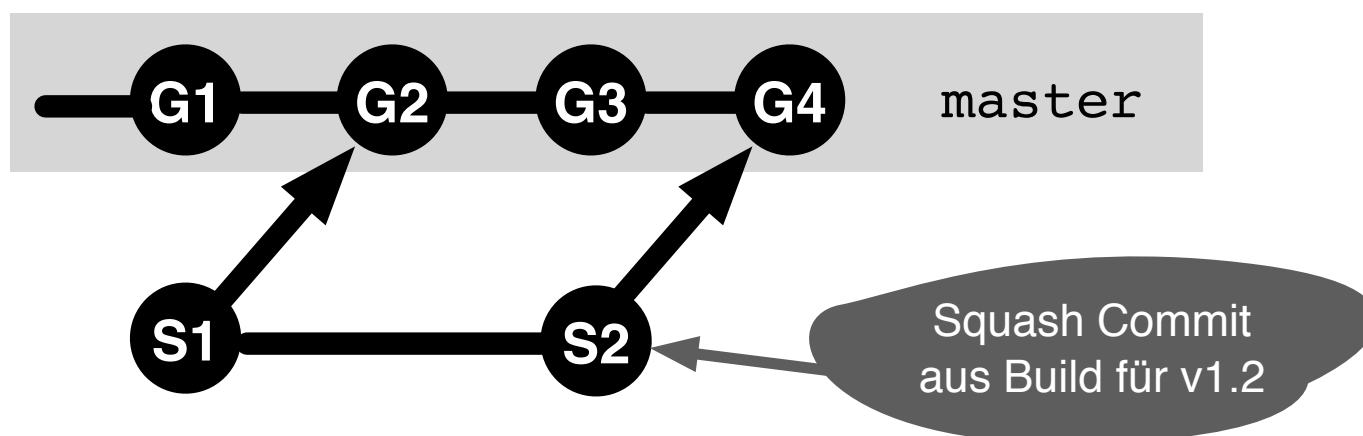
Neue Version festlegen



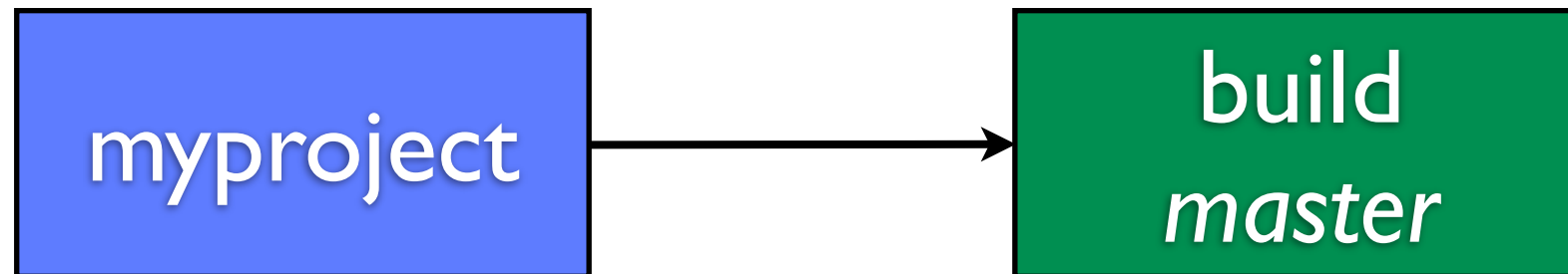
```
myproject$ git subtree pull --prefix build --squash ../build.git v1.2  
myproject$ git push
```



mit --squash

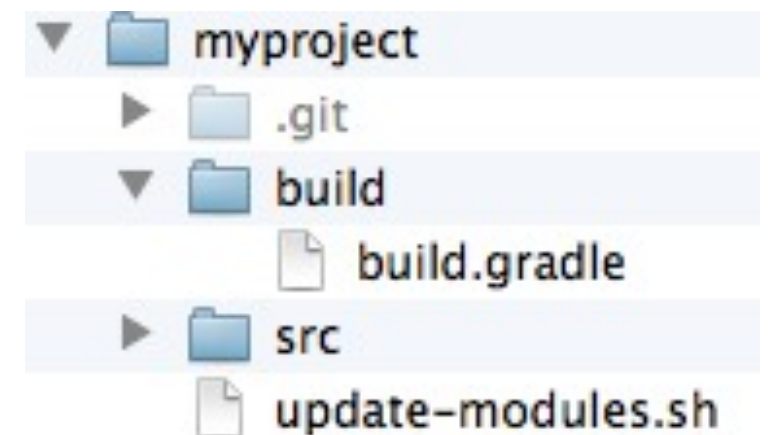


Aktuellste Version holen

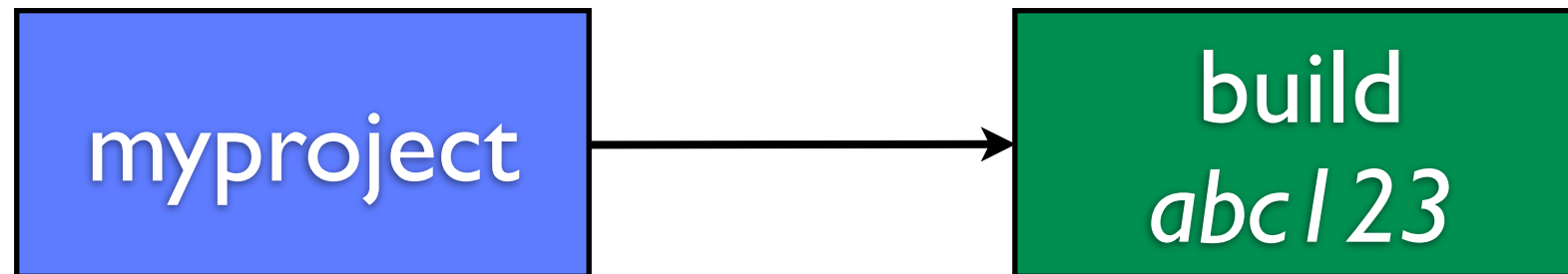


```
myproject$ git subtree pull --prefix build --squash ../build.git master
myproject$ git push
```

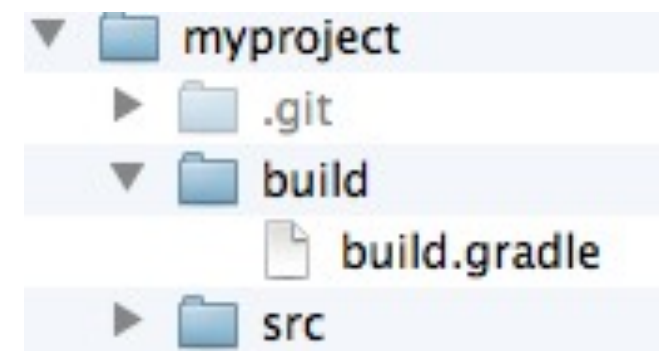
```
#Eigenes Update-Skript anlegen: update-modules.sh
myproject$ ./update-modules.sh
```



In Subtrees arbeiten

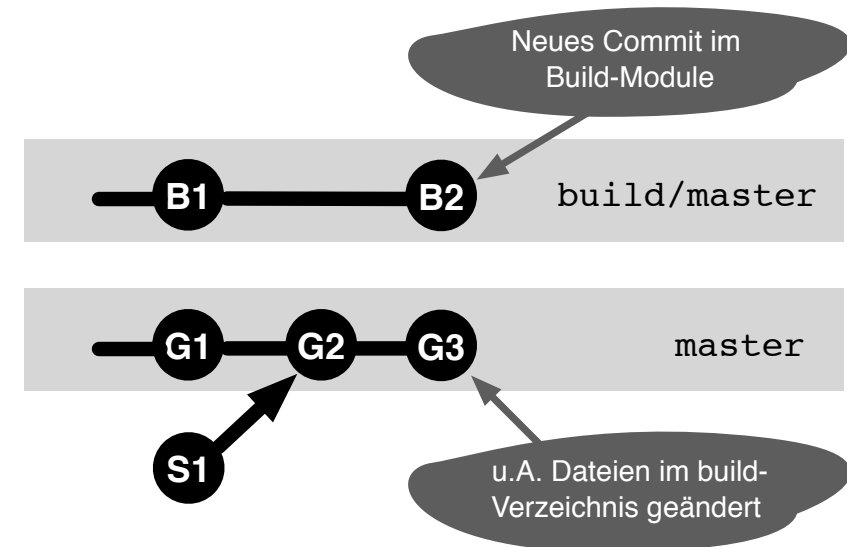


#Dateien im build-Verzeichnis ändern
myproject\$ git commit -am "neue Version in Subtree"
myproject\$ git push

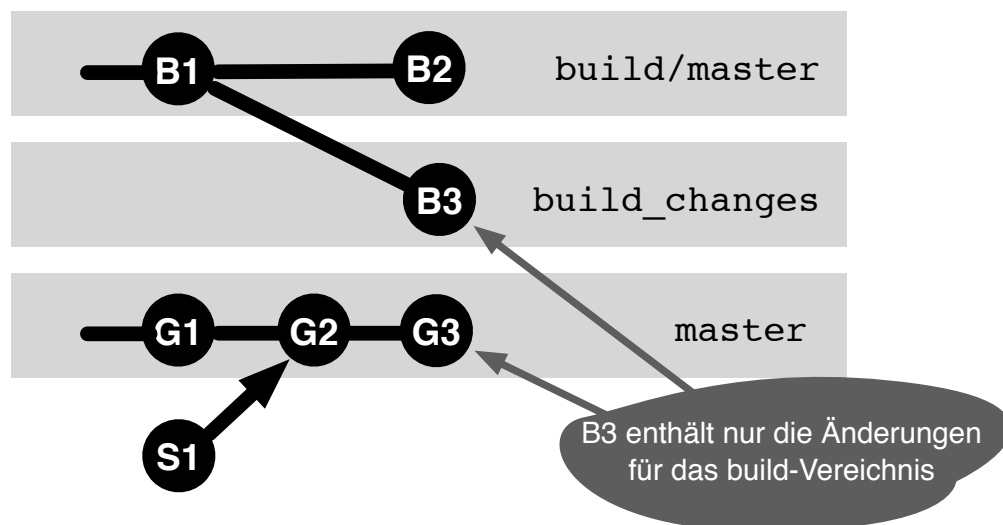


Änderungen splitten

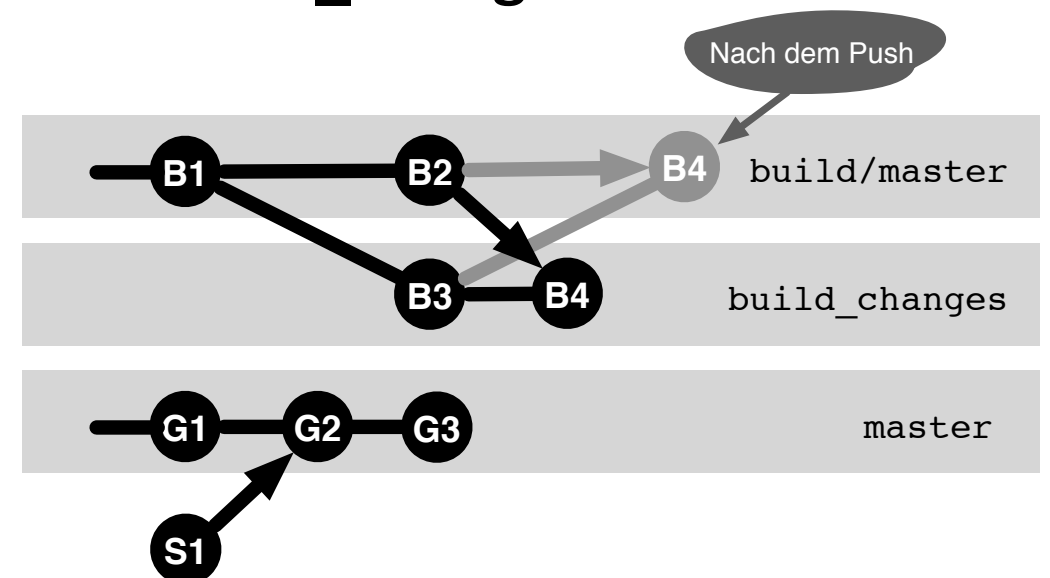
```
git remote add -f build ../build.git
```



```
git subtree split --prefix build --branch build_changes
```



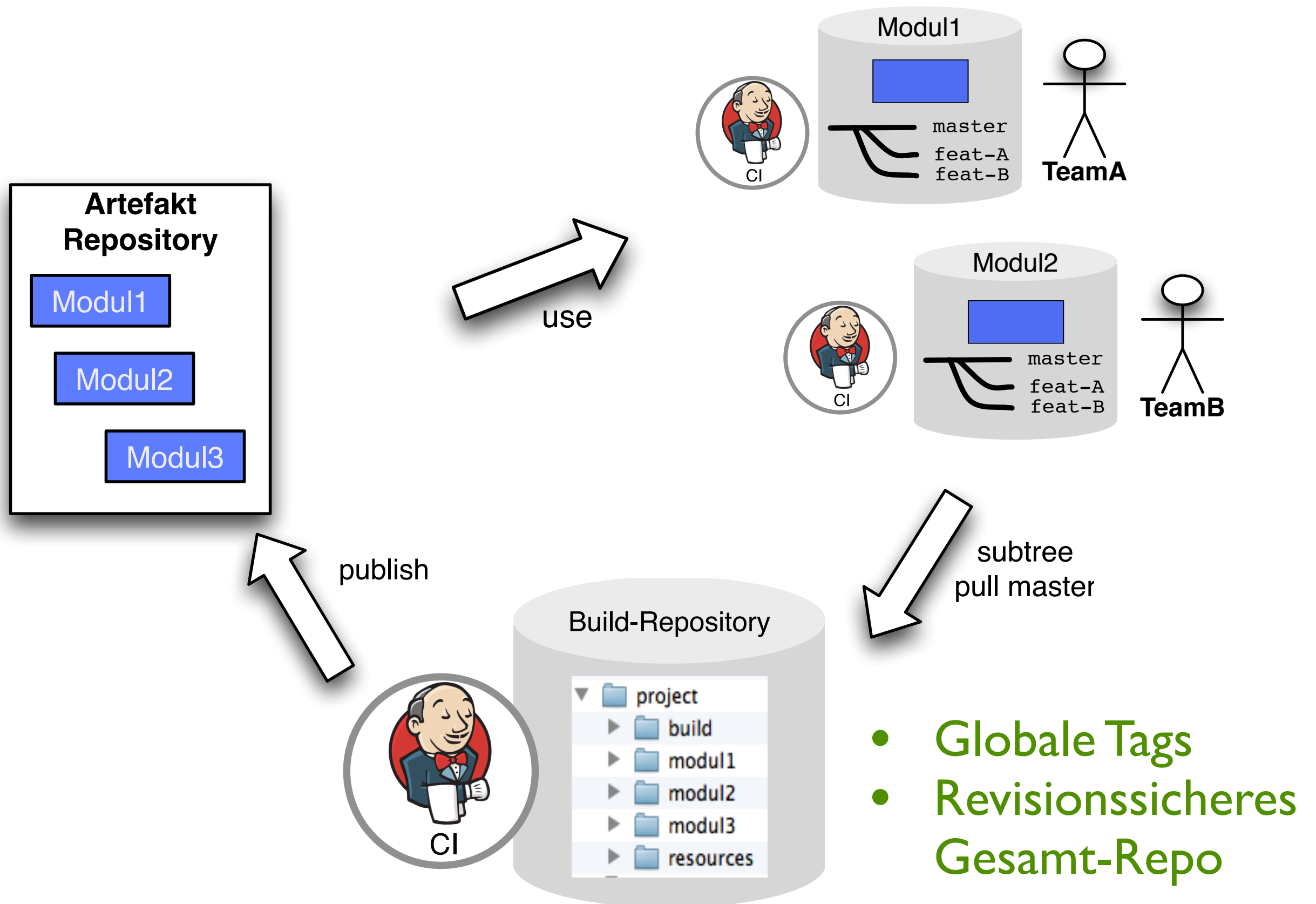
Merge+Push



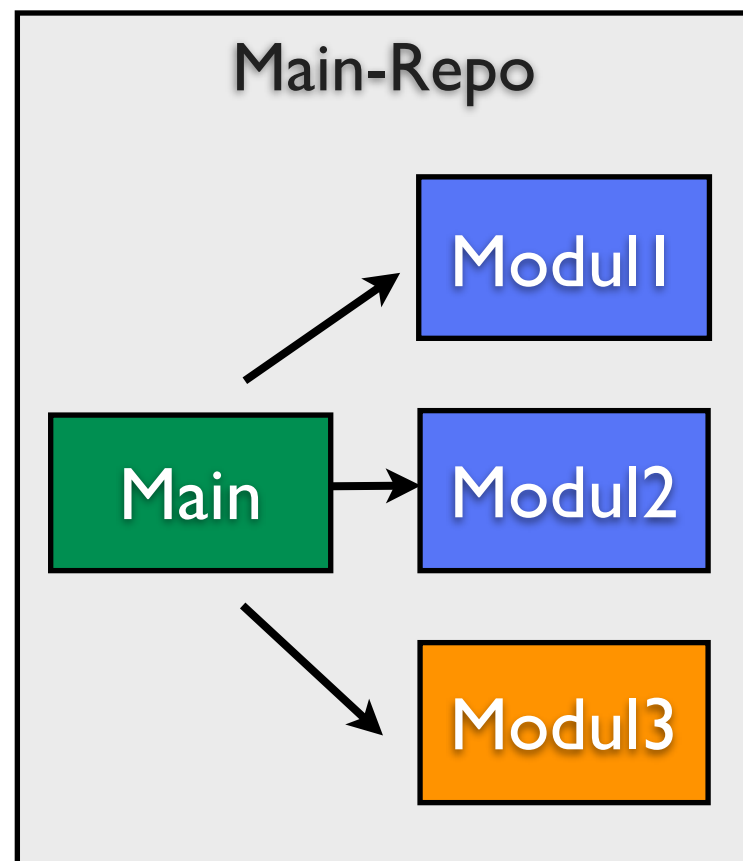
Subtree vs. Submodule

- Die Verwendung von Subtrees ist weniger komplex als die Verwendung von Submodulen.
- Die Restkomplexität trifft nur denjenigen, der auch direkt mit den Modulen arbeitet.
- Ein Repository enthält alle Dateien.
- Bei Subtrees ist die Historie der Dateien nicht im Hauptrepository einsehbar (--squash).

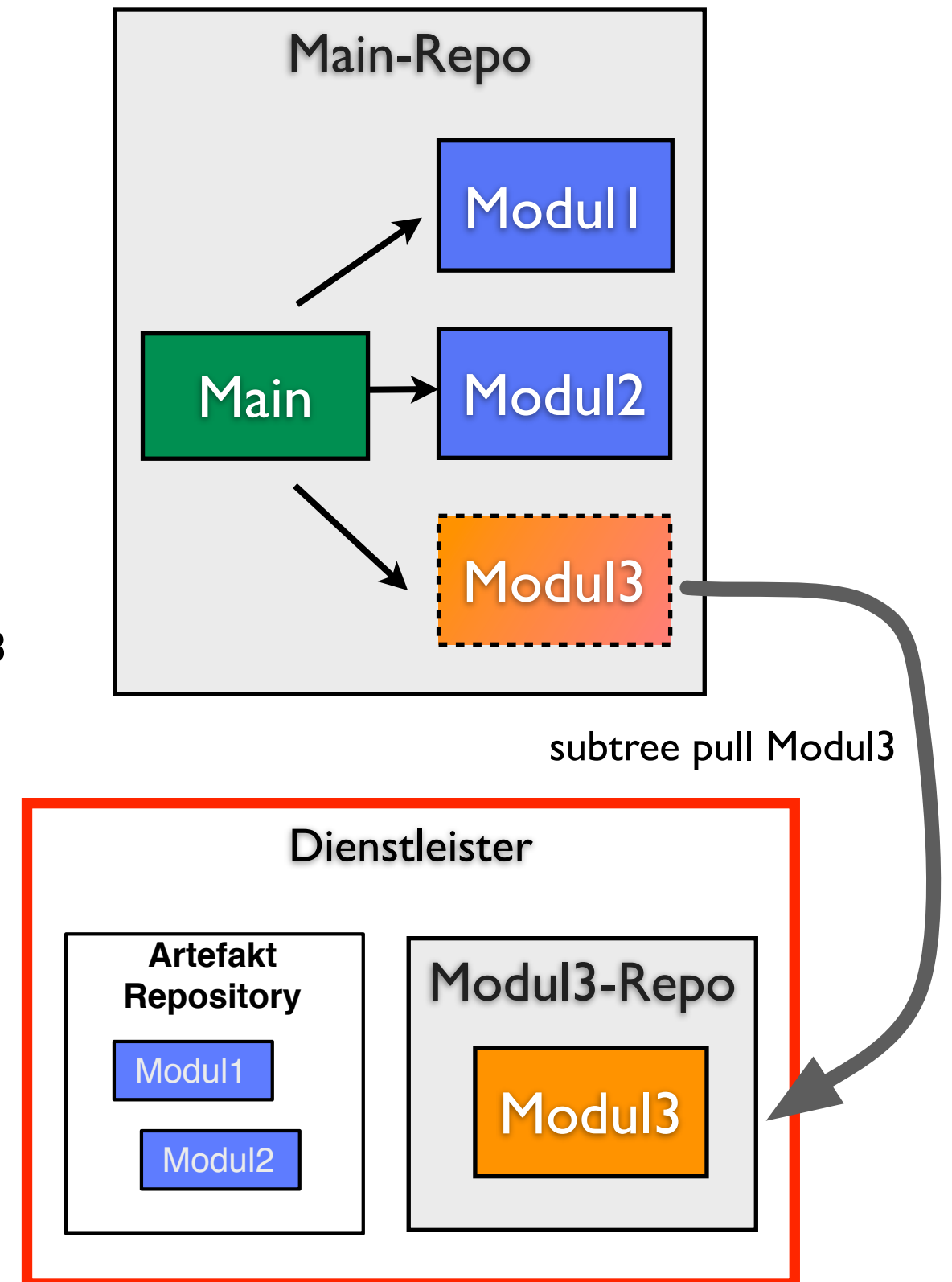
Beispiel - Globales Build

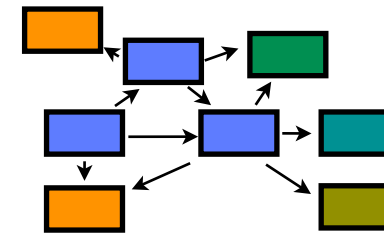
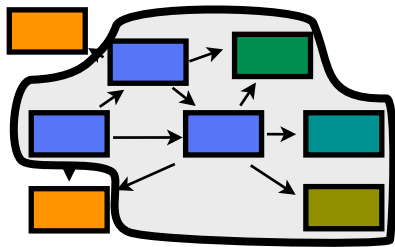


Beispiel: Externer Dienstleister



subtree split Modul3

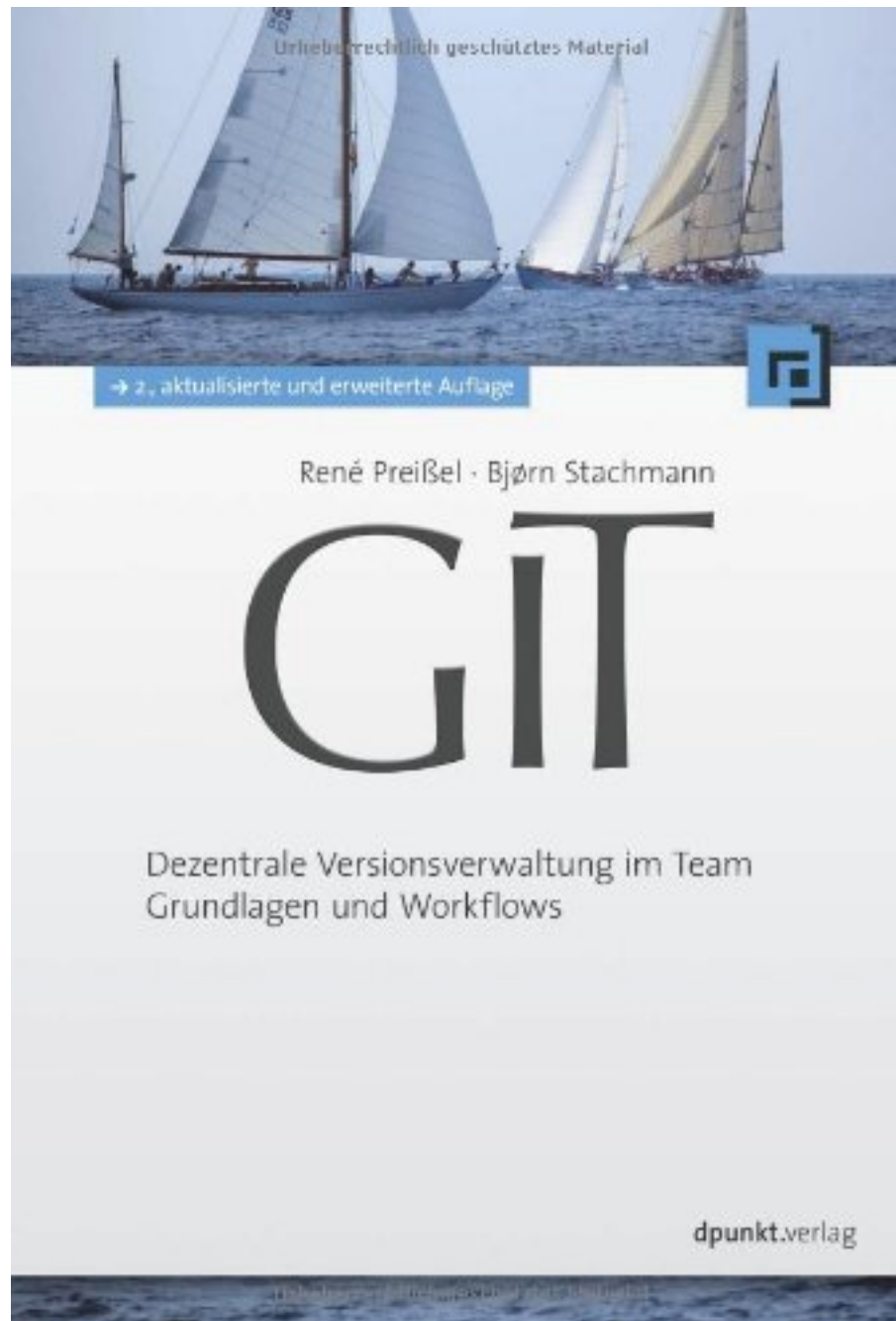




Wann was nutzen?

Projektstart - Ein Team	Ein Repo
Projektstart - Mehrere Teams	Ein Repo ggf. mit Team-Repos
Globales Build	Ein Repo ggf. mit Modul-Repos
Globale Features oder Refactorings	Ein Repo ggf. mit Modul-Repos
Module mit eigenen Releasezyklus	Dependency Manager falls vorhanden, sonst Subtree
Weitergabe einzelner Module	Splitten in eigenes Repo Einbinden mit Dependency Manager oder Subtree
Einbinden von externen Bibliotheken	Dependency Manager
Einbinden von Fremdsourcen	Subtree

Über Uns



René Preißel
@RenePreissel

Freiberuflicher
Berater, Entwickler, Trainer

Bjørn Stachmann
@old_stachi

Senior Software Engineer
etracker GmbH