



Pre-tested commit 2.0 mit Gerrit und Jenkins

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Steffen Schäfer

Steffen Schluff

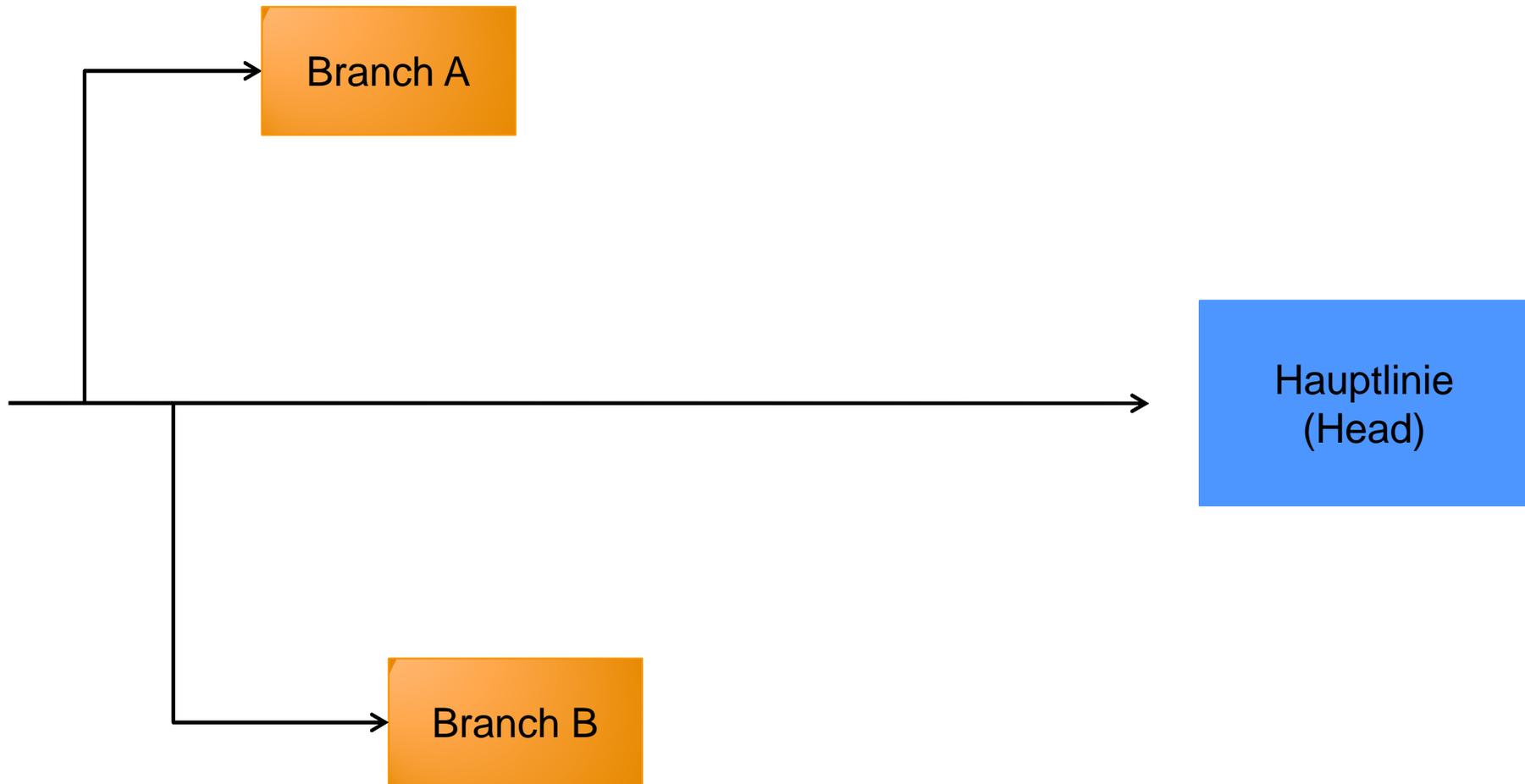
Gliederung



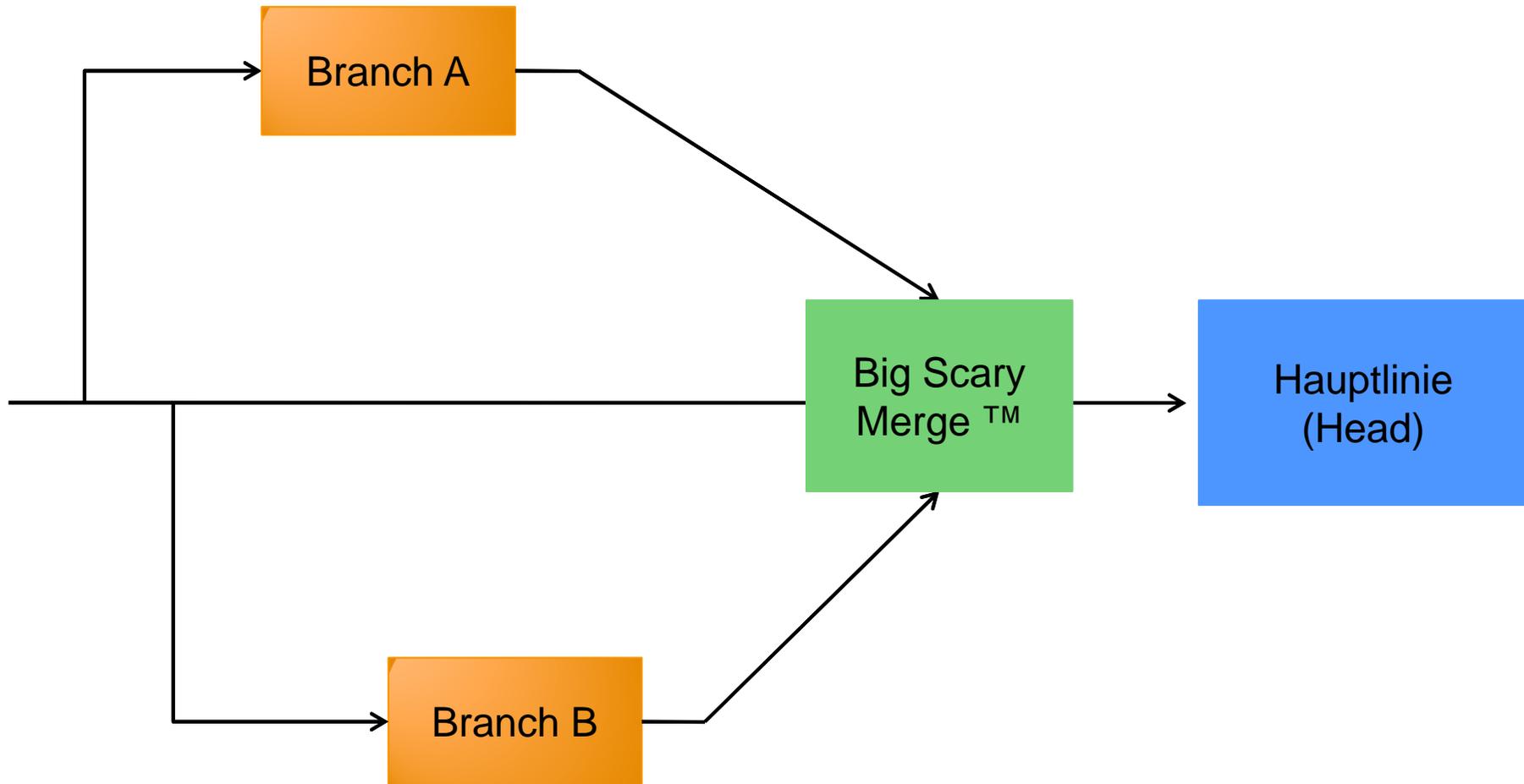
- Pre-tested commit
- Gerrit und Jenkins
- Pre-tested commit mit Gerrit und Jenkins

- Klassischer Wunsch: Isolation größerer Entwicklungsaufgaben
 - ... mit Gewährleistung einer sauberen Hauptentwicklungslinie
- „The key which opens real [...] parallel development are branches.“
 - Branch ist ein sich abspaltender neuer Entwicklungszweig
- Verteilte Versionsierungstools (DVCS) erleichtern Branches deutlich
- Branches sind Grundlage der meisten DVCS Workflows

Splendid isolation (2)



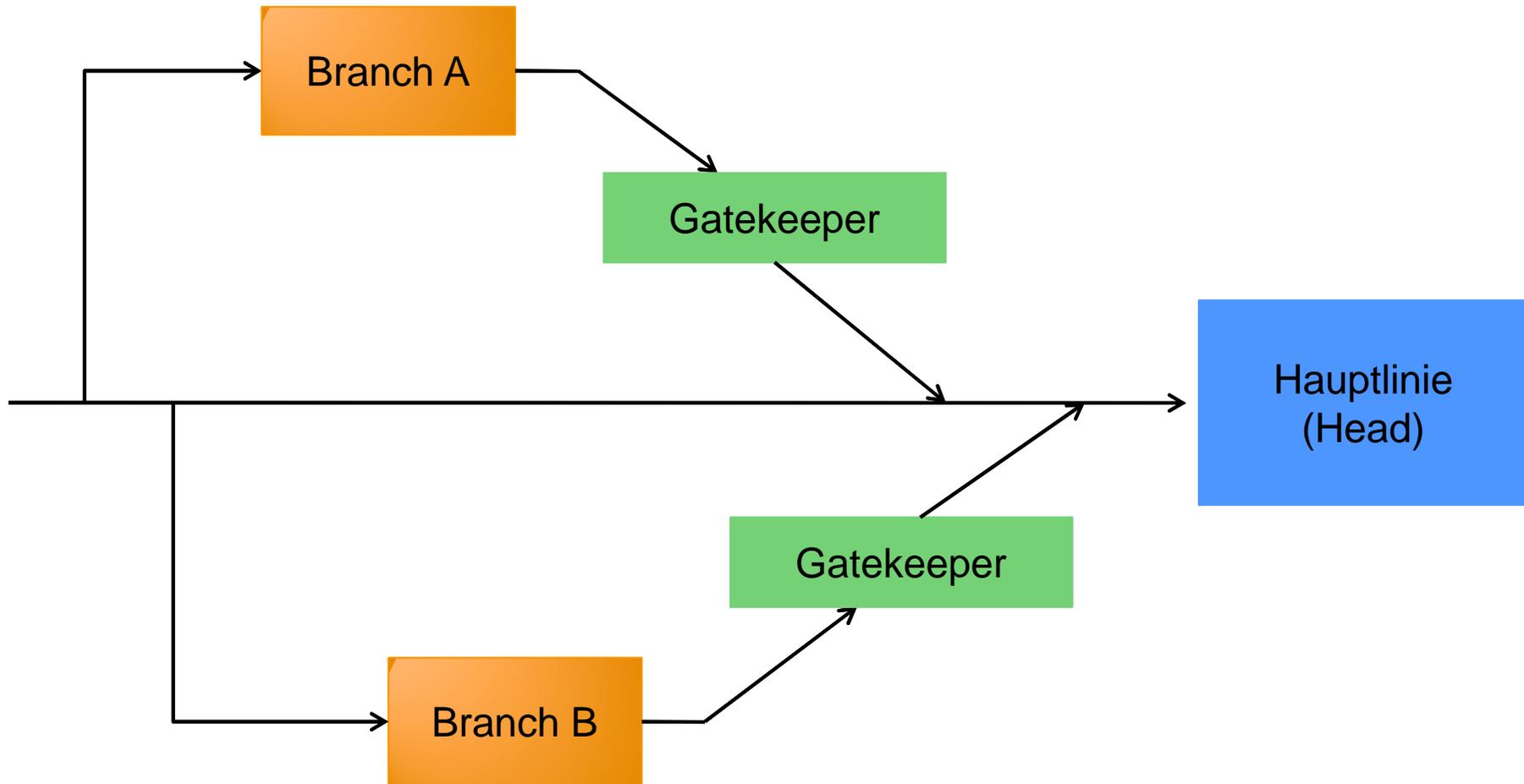
Here be dragons (1)



Here be dragons (2)

- Merge ist das erneute Zusammenführen von Branches
- Branches können sich textuell und semantisch unterscheiden
- „It's [...] particularly semantic conflicts that make big merges scary.“
- Stabilität der Hauptentwicklungslinie darf nicht leiden

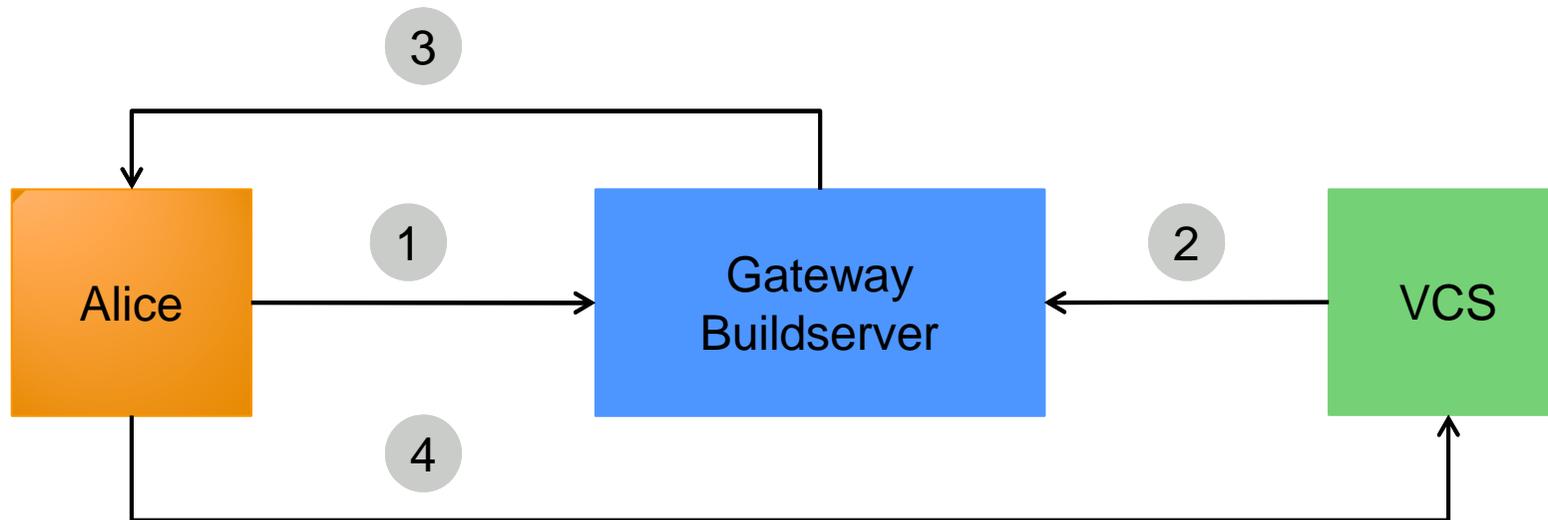
Are you the Keymaster? (1)



Are you the Keymaster? (2)



- Gatekeeper garantiert Stabilität des Heads
 - Bezeichnung kann variieren (z.B. Integrator)
 - Besitzt als einziger Schreibrechte in der Hauptentwicklungslinie
- Gatekeeper kann Mensch oder Maschine sein
 - Oder beides: Mensch prüft erst nach Maschine
 - Auch mehrere Menschen möglich: Peer Code Review einer Gruppe
- Wunsch „Saubere Hauptentwicklungslinie“ ist nicht DVCS spezifisch
 - Änderungen vor Commit als „persönlichen“ Gateway Build prüfen
 - In diversen CI Servern bereits verfügbar („pre-tested commit“)



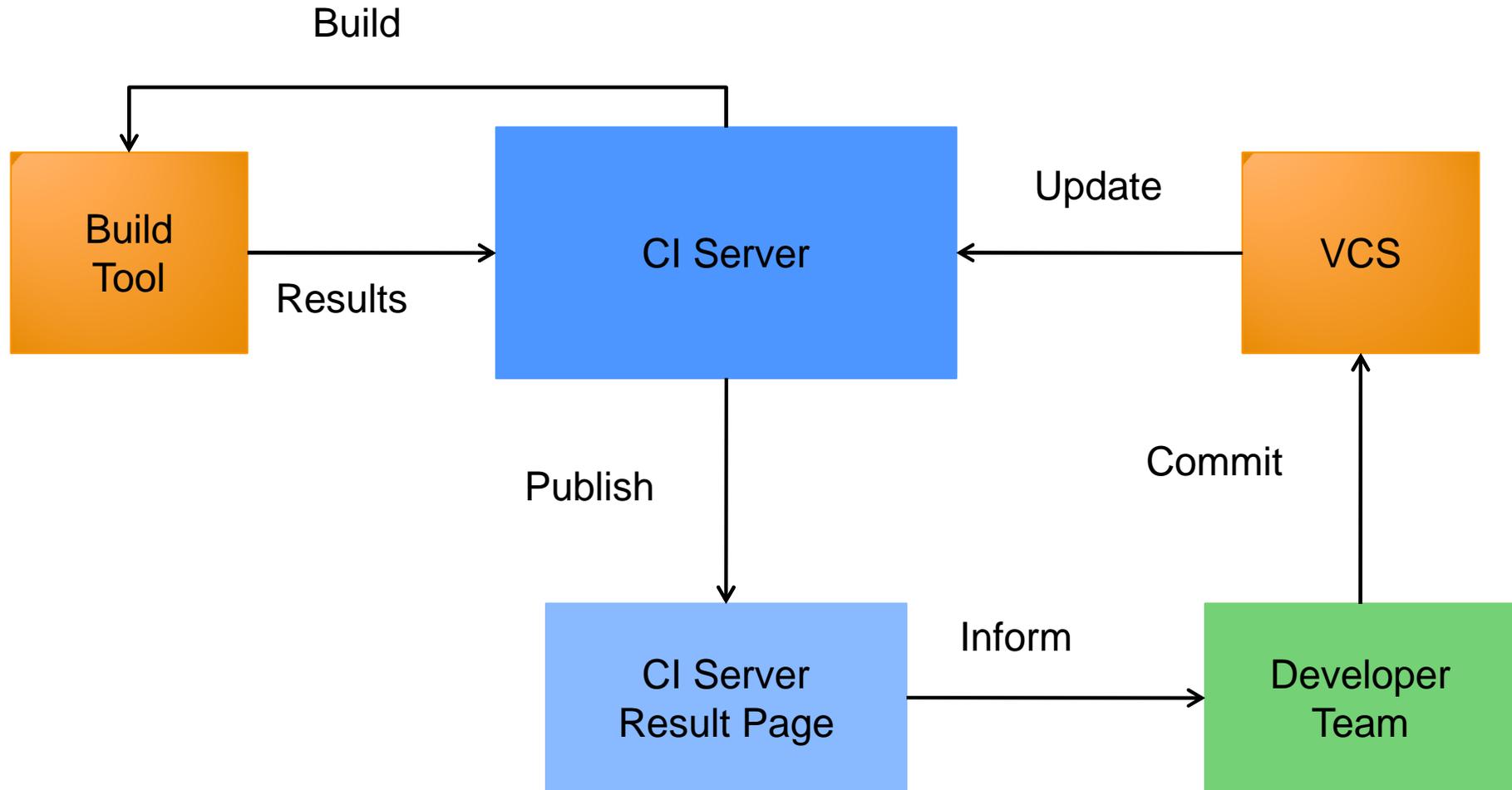
- Benutzer schickt Änderungen zum Testen an Gateway
- Gateway testet VCS Stand mit Änderungen
- Gateway sendet Testergebnis an Benutzer
- Commit der Änderungen durch Benutzer wenn Testergebnis OK

Gliederung



- Pre-tested commit
- **Gerrit und Jenkins**
- Pre-tested commit mit Gerrit und Jenkins

Been there, done that



Jenkins, at your service

- CI-Server
- Open Source
- Durch Plugins erweiterbar
- Vielzahl vorhandener Plugins

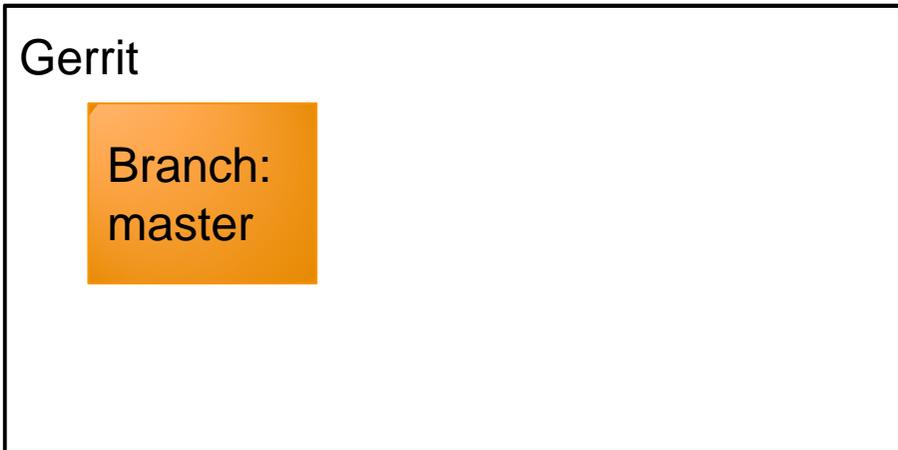


Quelle: <http://jenkins-ci.org>

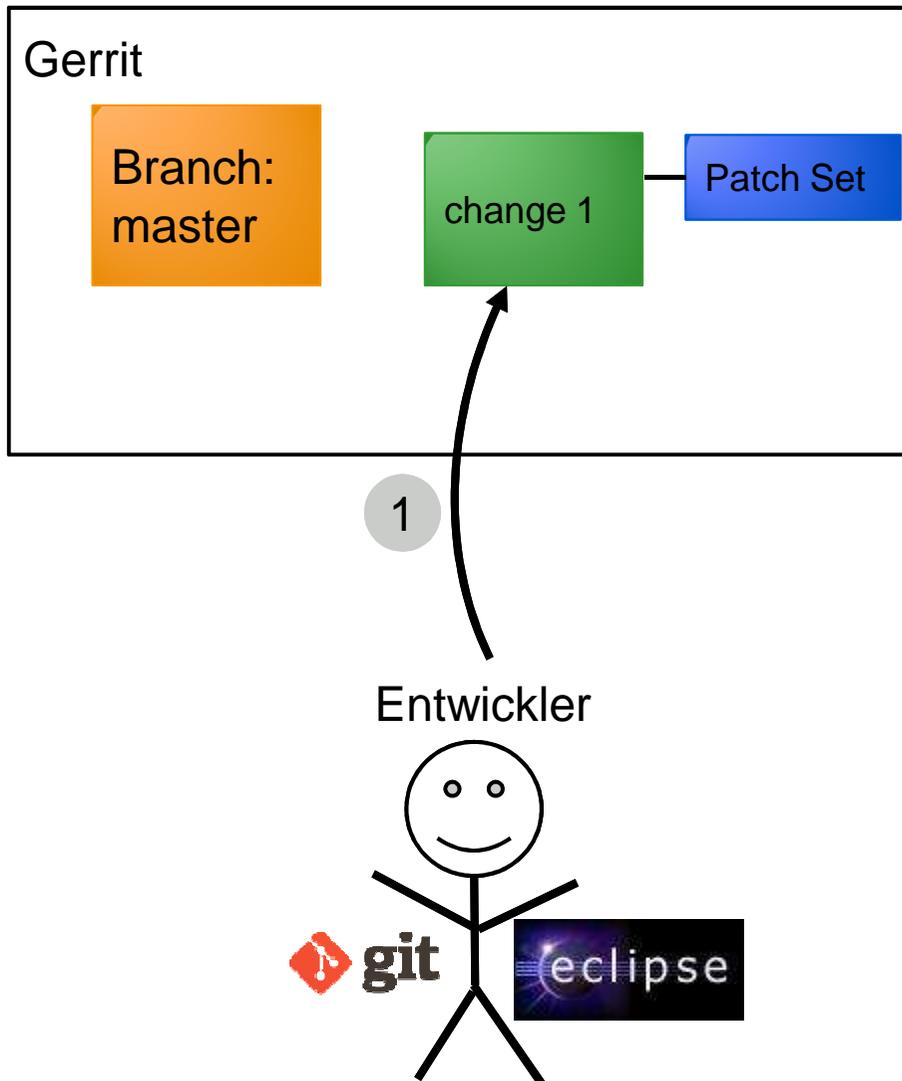
- Code Review Werkzeug für Git
 - Entstand im Kontext des Android Projektes
 - Wurzeln liegen in Rietveld (SVN Code Reviews)
- Review der Änderungen vor Integration in den Entwicklungszweig
 - Stärke gegenüber anderen Tools
- Fungiert selbst als Git Server
 - Transparent für den Client
 - Bekanntes Client Tooling nutzbar
 - Rechtekonzept(!)



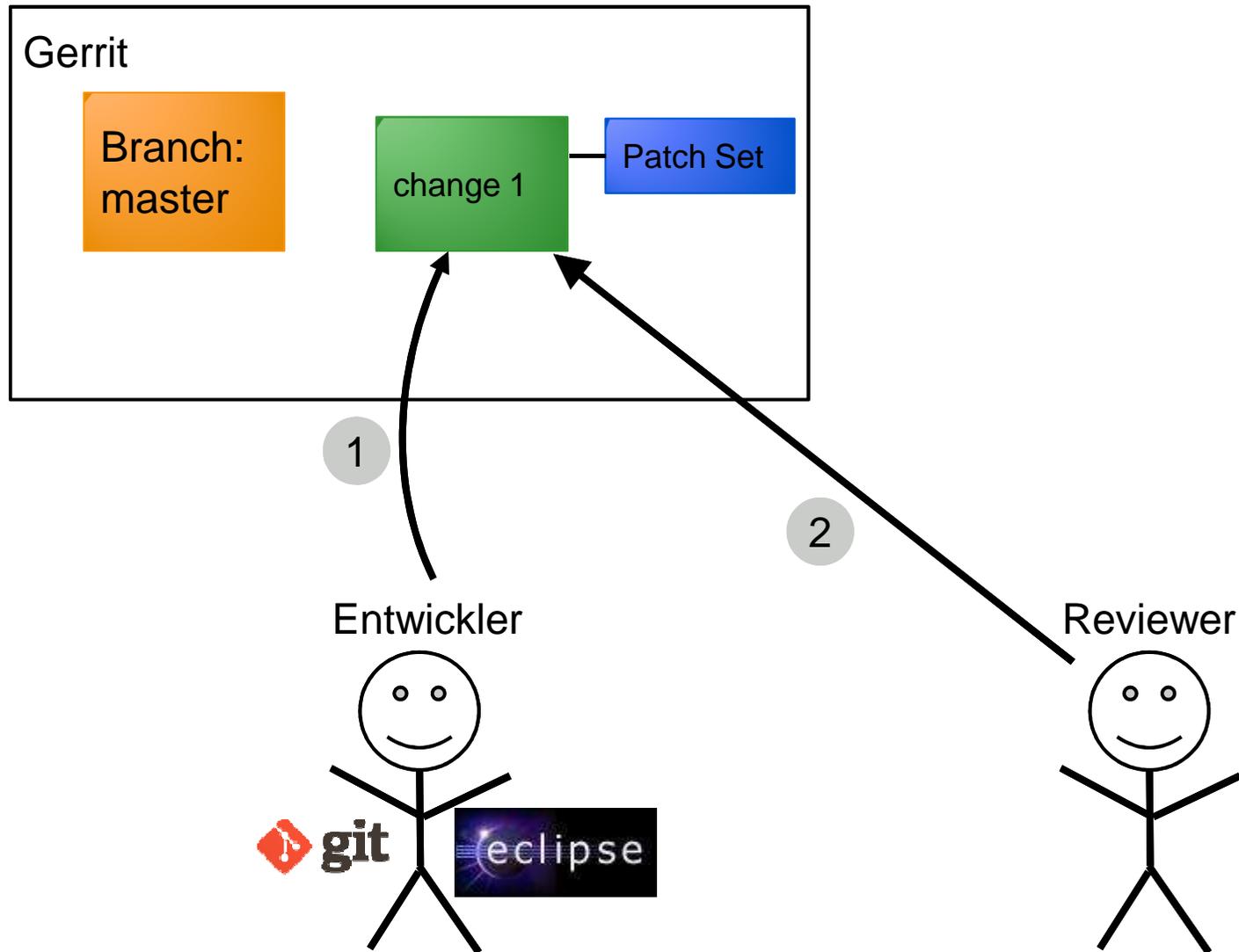
Gerrit - Workflow



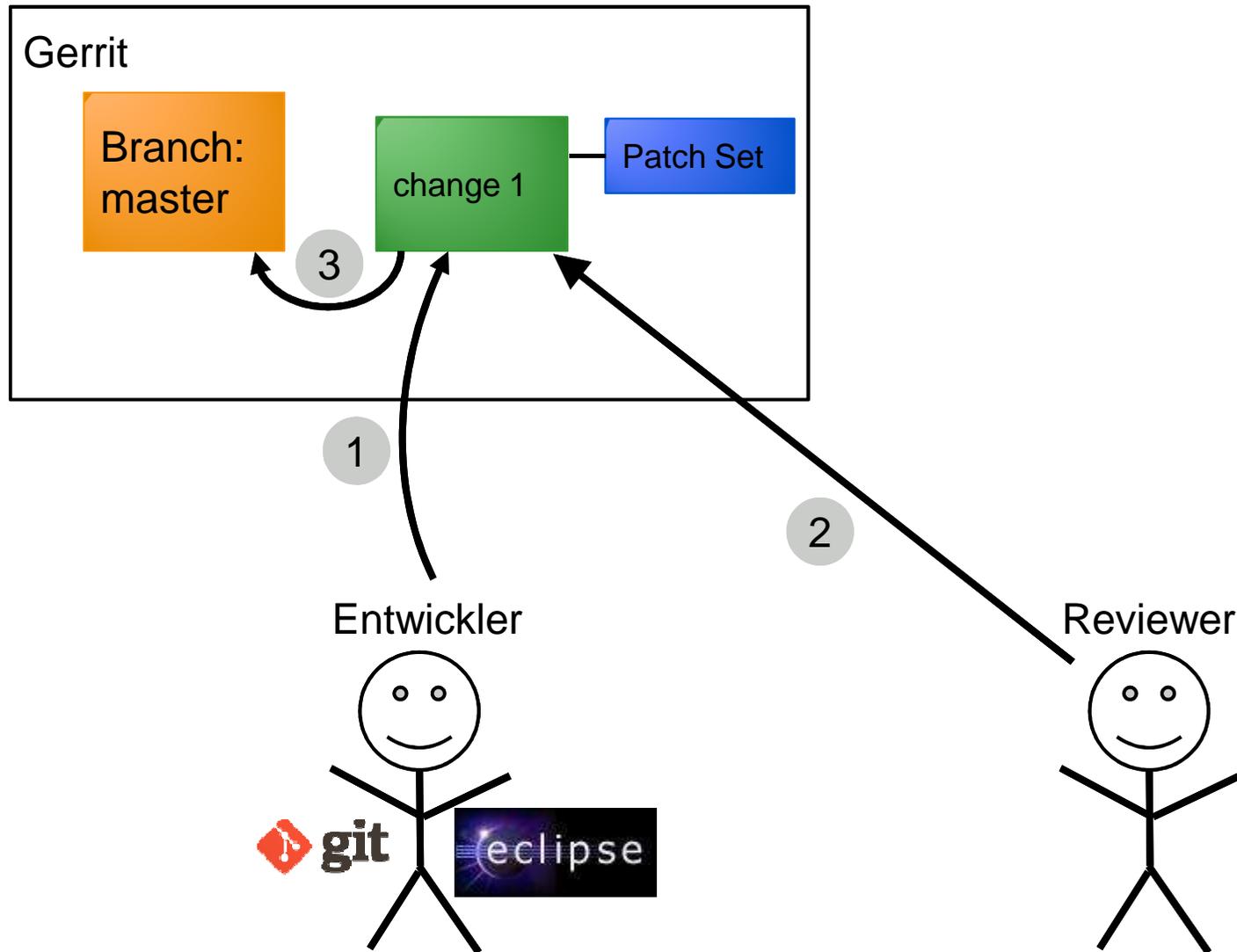
Gerrit - Workflow



Gerrit - Workflow

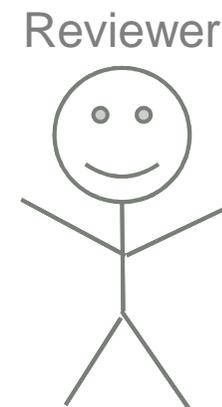
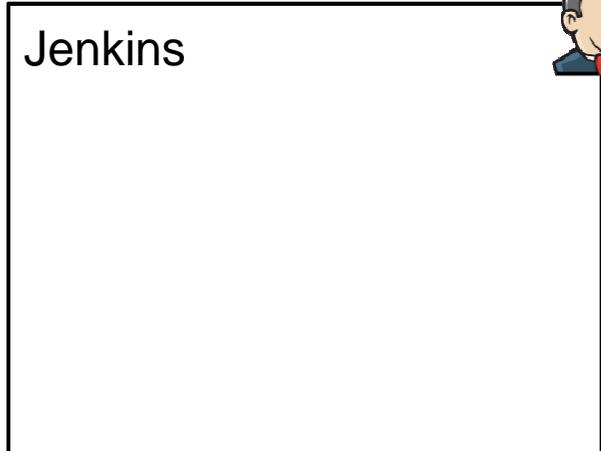
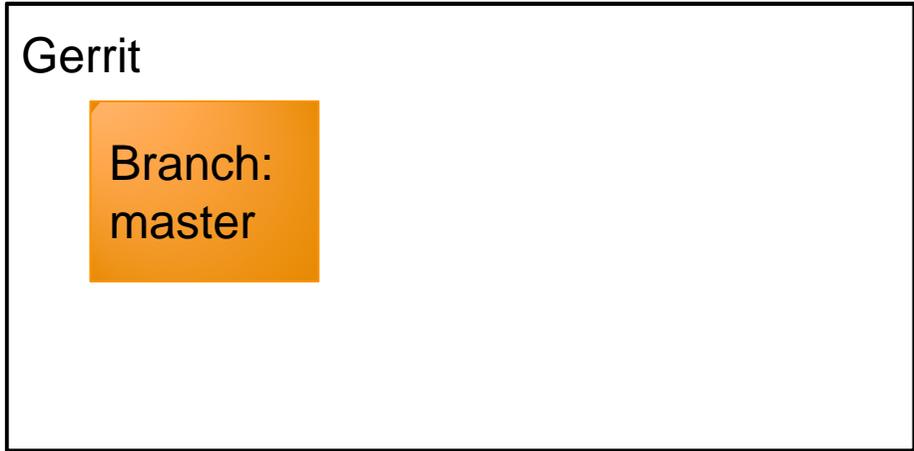


Gerrit - Workflow

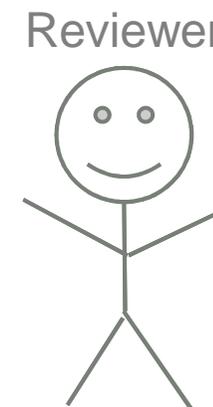
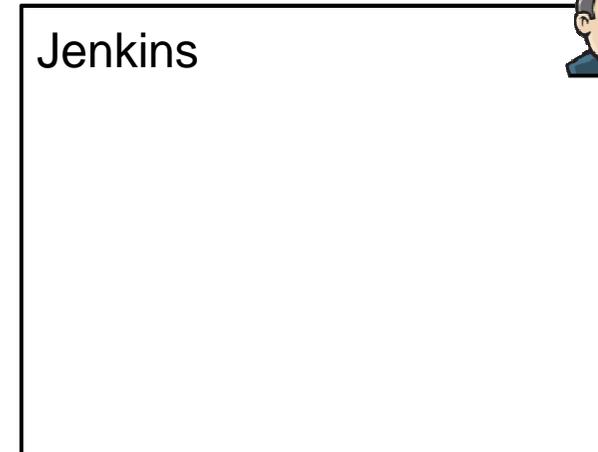
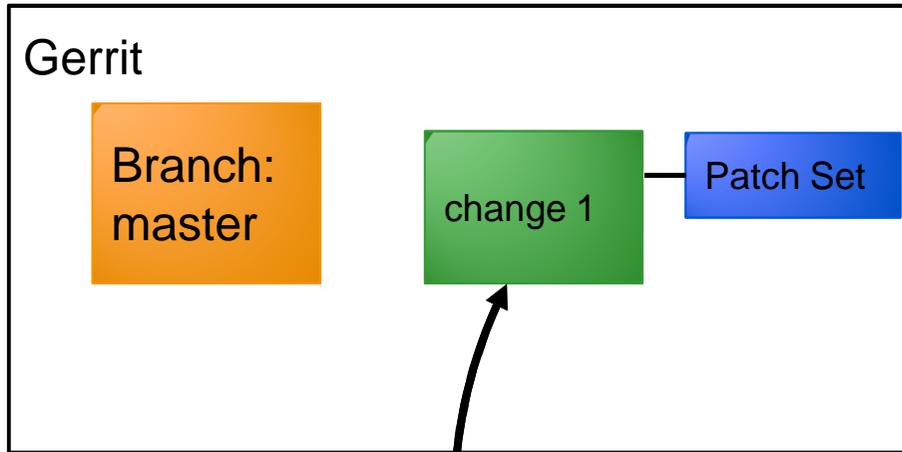


- Jenkins Plugin
- Beinhaltet spezifische Erweiterungen für Gerrit
- Neuer Build Trigger für Gerrit
 - Erlaubt das Anstoßen von Builds bei Erzeugung von Patch Sets
- Erweitert Konfiguration für Git Repo Zugriff
 - Ermöglicht einen Build auf Basis von Patch Sets
- Gibt nach dem Build Feedback zum Change
- Summe: Jobs zum Bauen von Gerrit Changes nun möglich

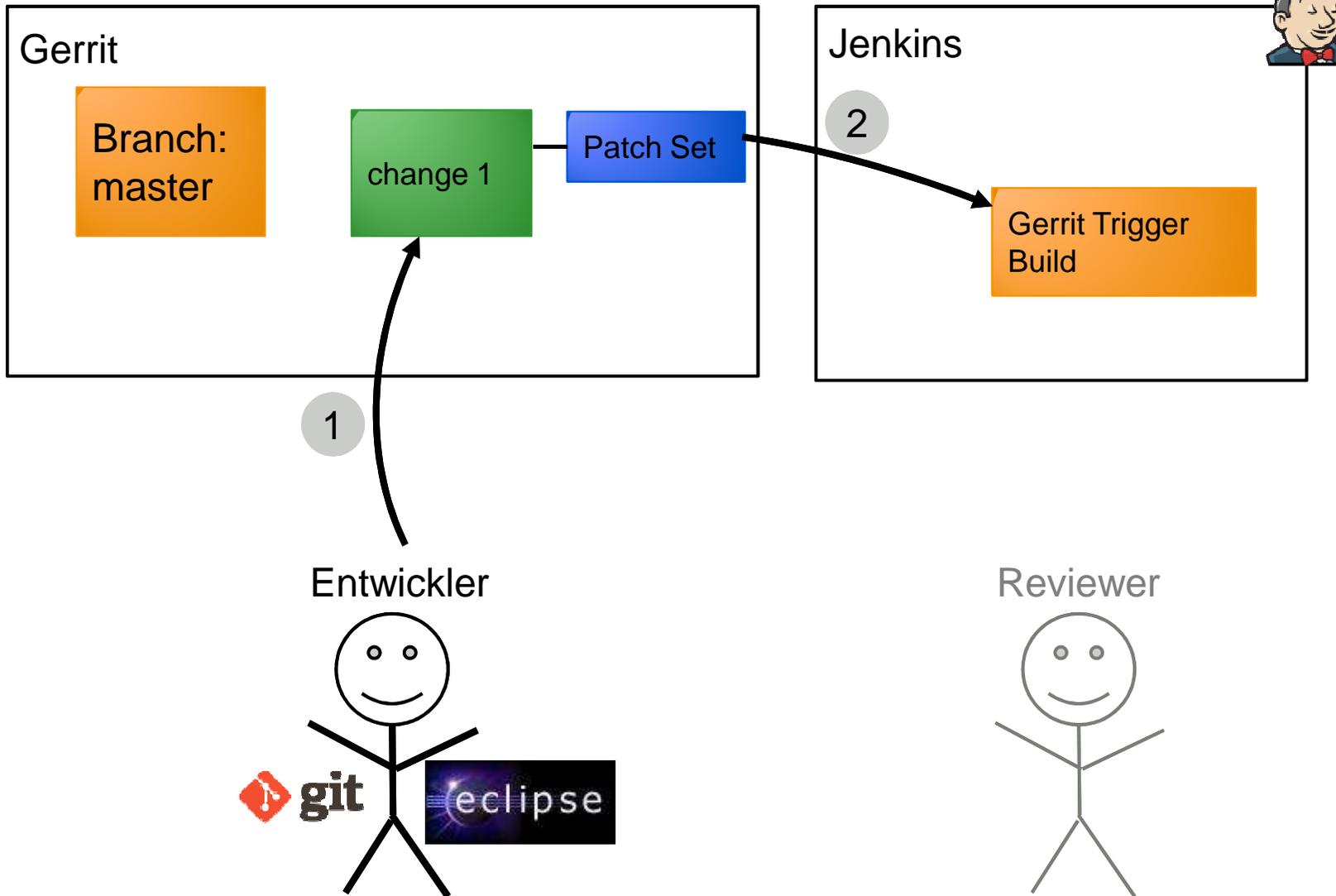
Gerrit + Jenkins - Workflow



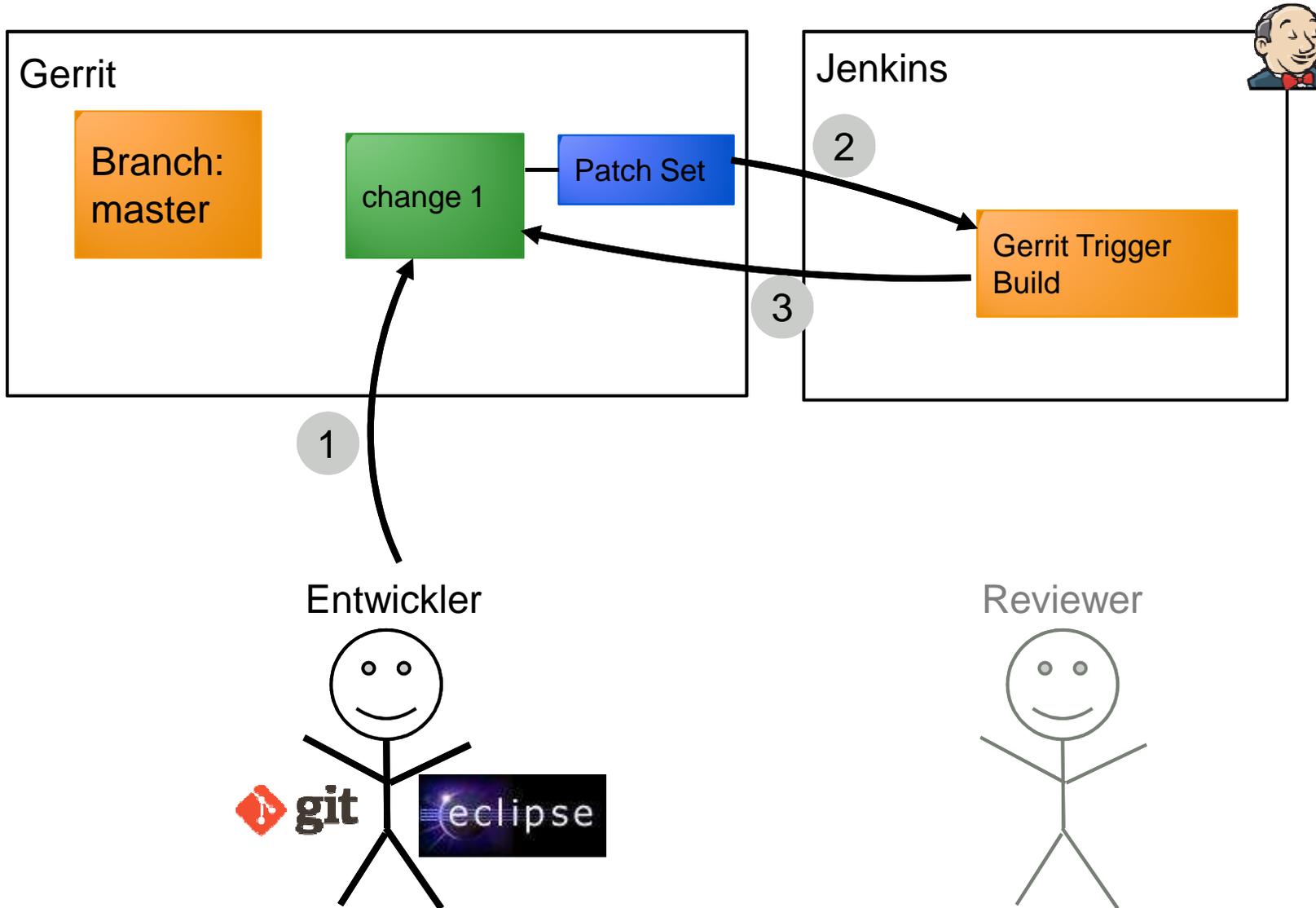
Gerrit + Jenkins - Workflow



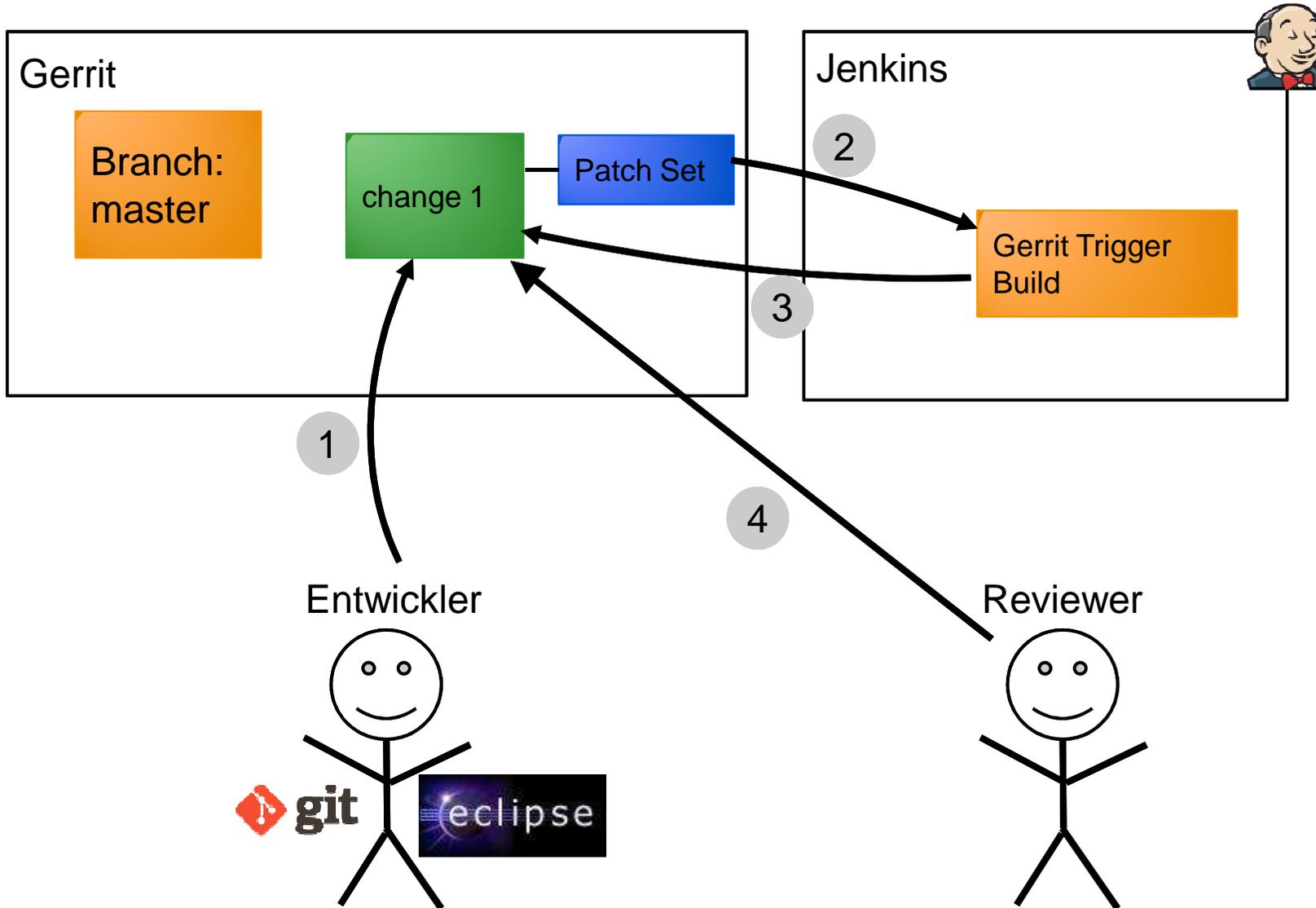
Gerrit + Jenkins - Workflow



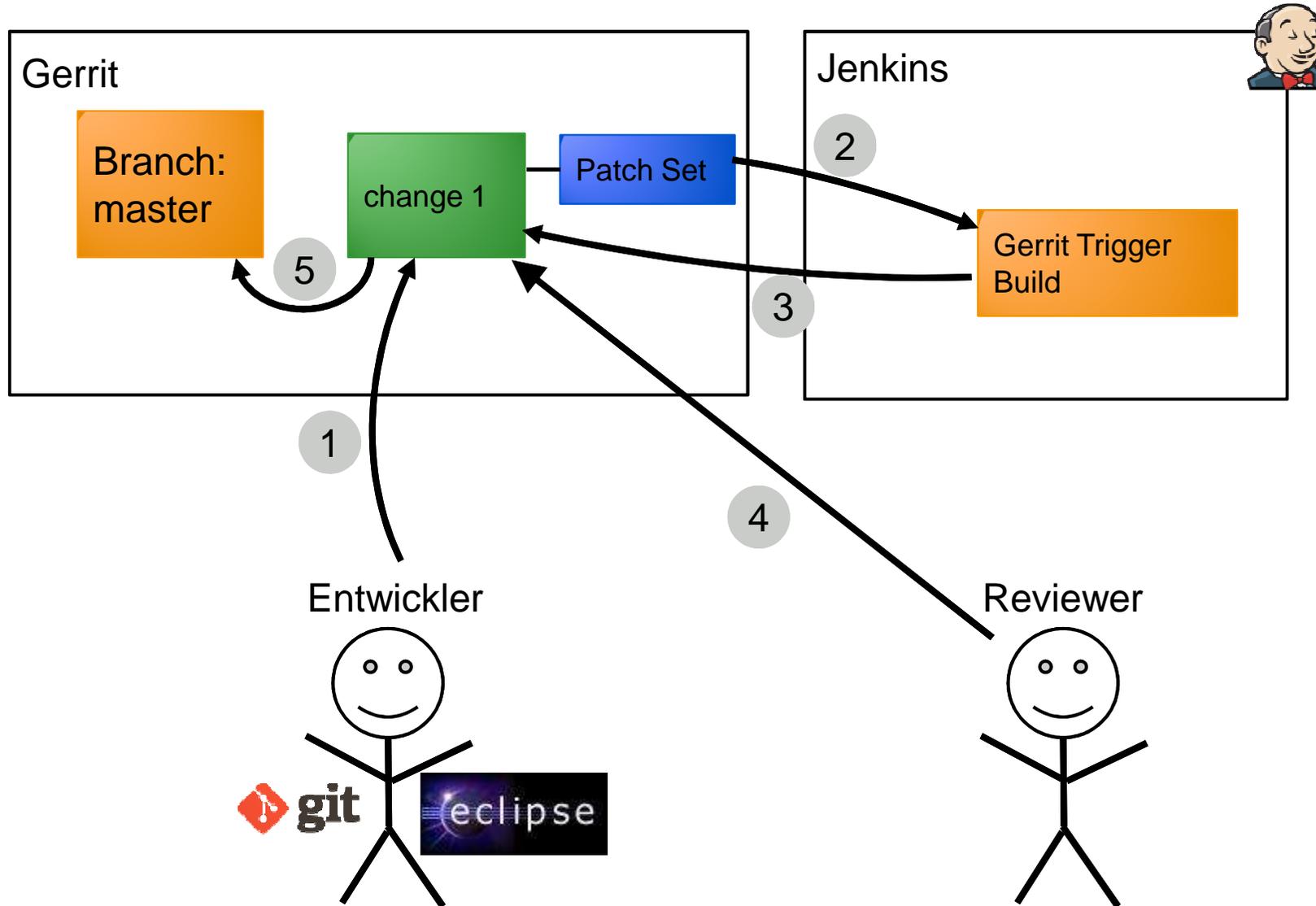
Gerrit + Jenkins - Workflow



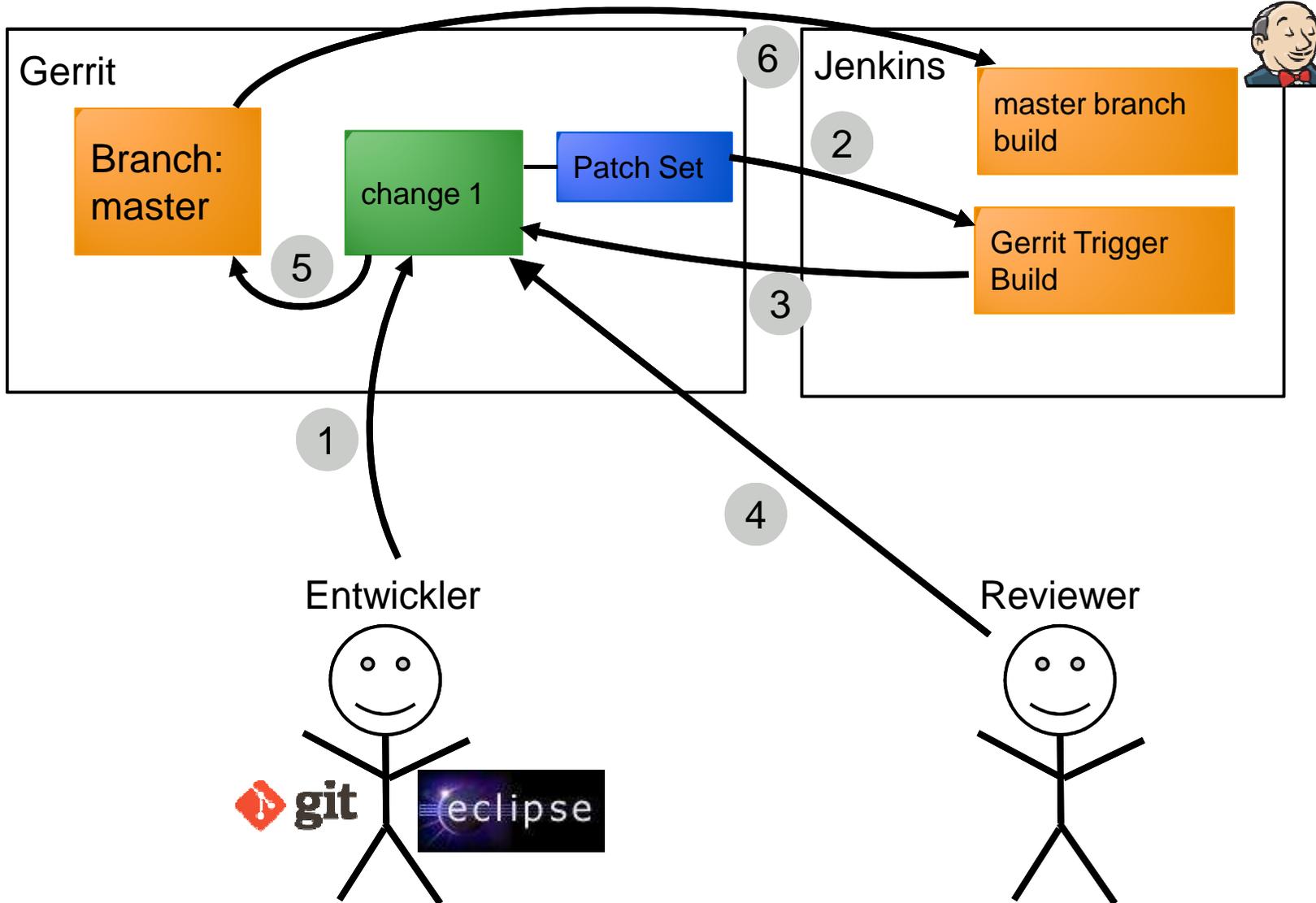
Gerrit + Jenkins - Workflow



Gerrit + Jenkins - Workflow



Gerrit + Jenkins - Workflow



Demonstration

- Erzeugung von Changes
- Build über Gerrit Trigger
- Integration von Changes in den Entwicklungszweig

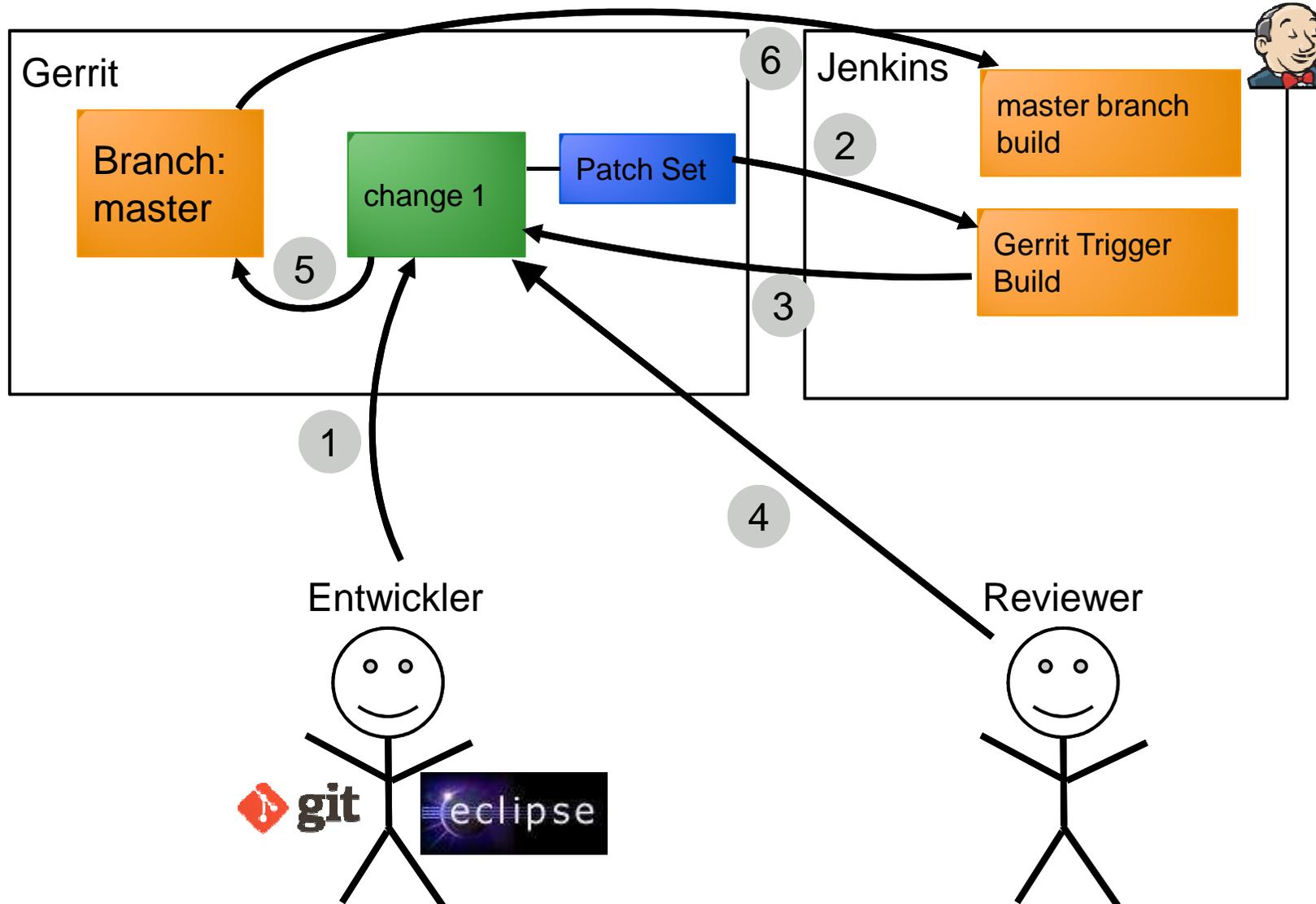


Gliederung

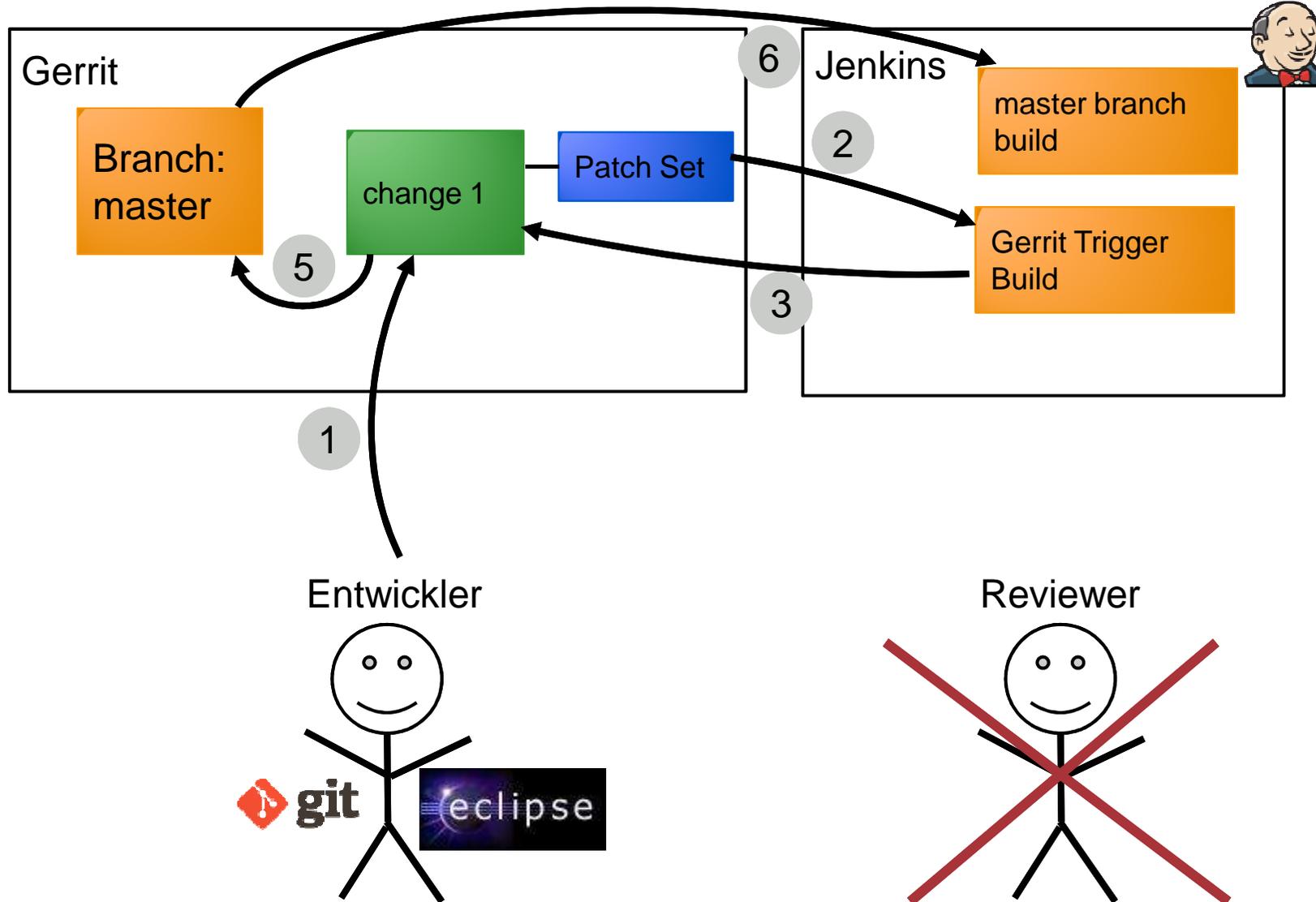


- Pre-tested commit
- Gerrit und Jenkins
- **Pre-tested commit mit Gerrit und Jenkins**

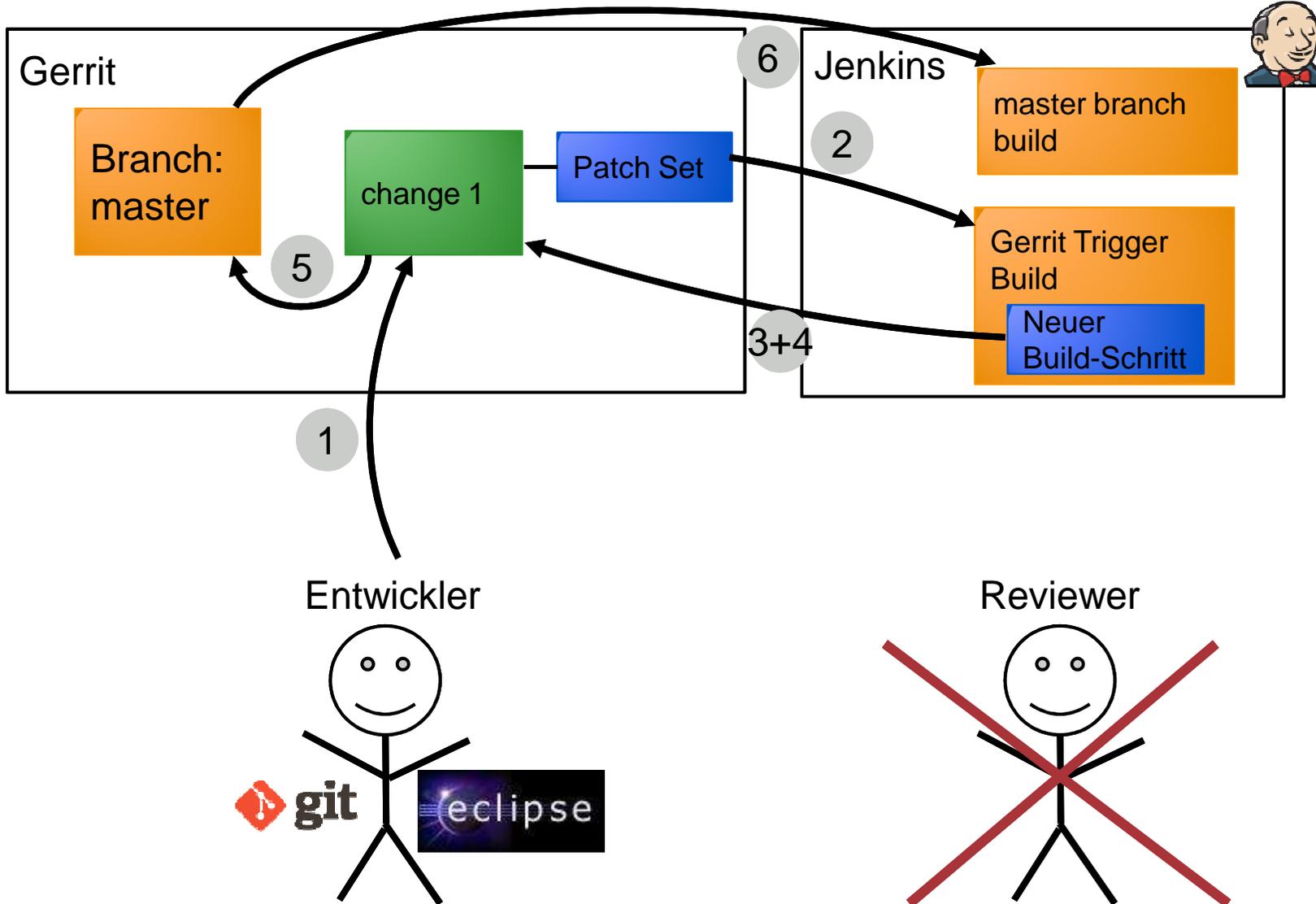
Wo ist jetzt der Pre-Tested commit?



Pre-tested commit mit Gerrit und Jenkins



Pre-tested commit mit Gerrit und Jenkins



Give me the grizzly details

- Weiterer Build Schritt
 - Typ: Shell
 - `ssh -p 29418 <GERRIT_USER>@<HOST> gerrit approve --verified +1 -code-review +2 --project <PROJECT_NAME> --submit $GERRIT_PATCHSET_REVISION`

Pre-tested commit - Wer macht das noch?



- IntelliJ + Teamcity
 - Vorreiter in diesem Bereich
 - Tool-abhängiger Ansatz
 - Abschließende Integration in das Repository erfolgt durch die IDE
 - http://www.jetbrains.com/teamcity/features/delayed_commit.html
- Bamboo 4.0
 - Gatekeeper für Feature Branches (Teil von Automatic Branch Merging)
 - <http://www.atlassian.com/software/bamboo/overview/dvcs>
- Hudson / Jenkins
 - Konzept im Wiki
 - <https://wiki.jenkins-ci.org/display/JENKINS/Designing+pre-tested+commit>

- Realisierung von Pre-tested commits leicht möglich
 - Rein Server-seitiger Workflow, keine Bindung an spezielle Clients
- Pre-tested commits für alle sichtbar
 - Und leicht lokal reproduzierbar (cherry-pick)
- Toolstack ist Open Source
 - Free speech and free beer
- Pretested commit auf Basis von Gerrit
 - Somit Code Review auf Wunsch nutzbar

- Continuous vs. Controlled Integration
 - <http://www.ddj.com/architect/205917960>
- MF Bliki: FeatureBranch
 - <http://martinfowler.com/bliki/FeatureBranch.html>
- Jenkins
 - <http://jenkins-ci.org/>
- Gerrit
 - <http://code.google.com/p/gerrit/>
- Gerrit Trigger
 - <https://wiki.jenkins-ci.org/display/JENKINS/Gerrit+Trigger>



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Vielen Dank für ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de