

Gitによる 多人数開発における 切り戻し方法

How to do rolling-back in Git used among multiple members

Vol.01 Aug/23/2013

Ryotaro Ikeda

Development Process Optimization Department, Rakuten Inc.

<http://www.rakuten.co.jp/>

自己紹介

```
/Users/ryotaro.ikeda > finger -l ryotaro.ikeda
```

Login: ryotaro.ikeda (池田 涼太郎)

Shell: /usr/local/bin/fish

Office: Shinagawa / Rakuten, Inc.

On since Apr, 2012

No Plan.

Background:

Android/iOS, API(JavaEE6)/サービス運用

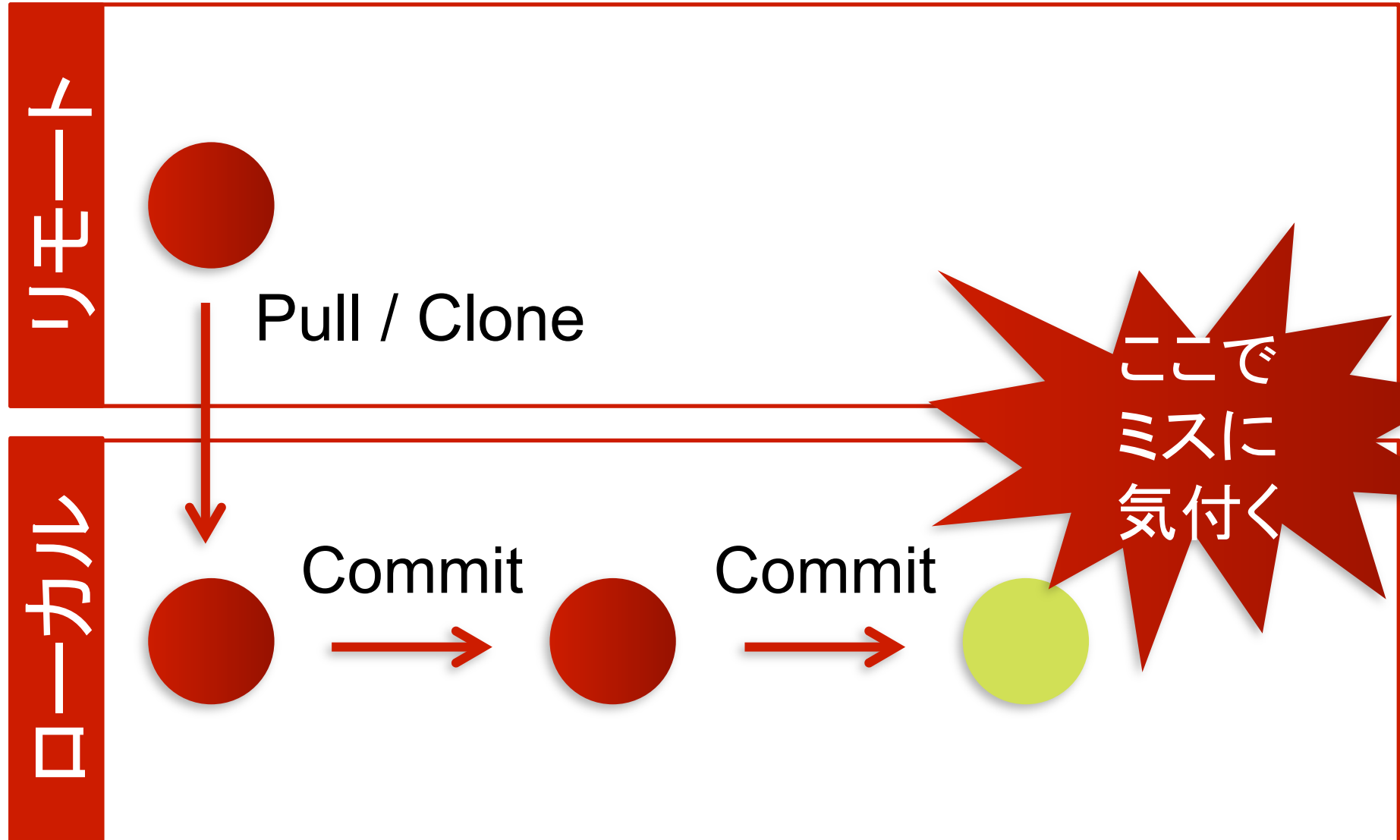
Chef/Docker等を用いたインフラ支援

本日お話すること

Gitで間違いコミットを リモートに作ってしまった時 どうするかのお話

切り戻しの シチュエーション

特に考えなくていい場合

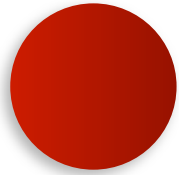


ローカルで間違えたときはどうという事はない

- リセットして戻せばいい
- オペミスが怖ければブランチ切れればいい
- それも怖ければzip -rすればいい

考えなきゃいけない場合

リモート

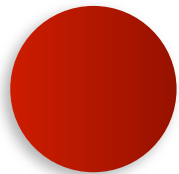


Pull / Clone

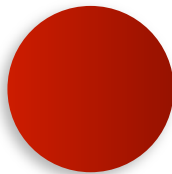


Push

ローカル



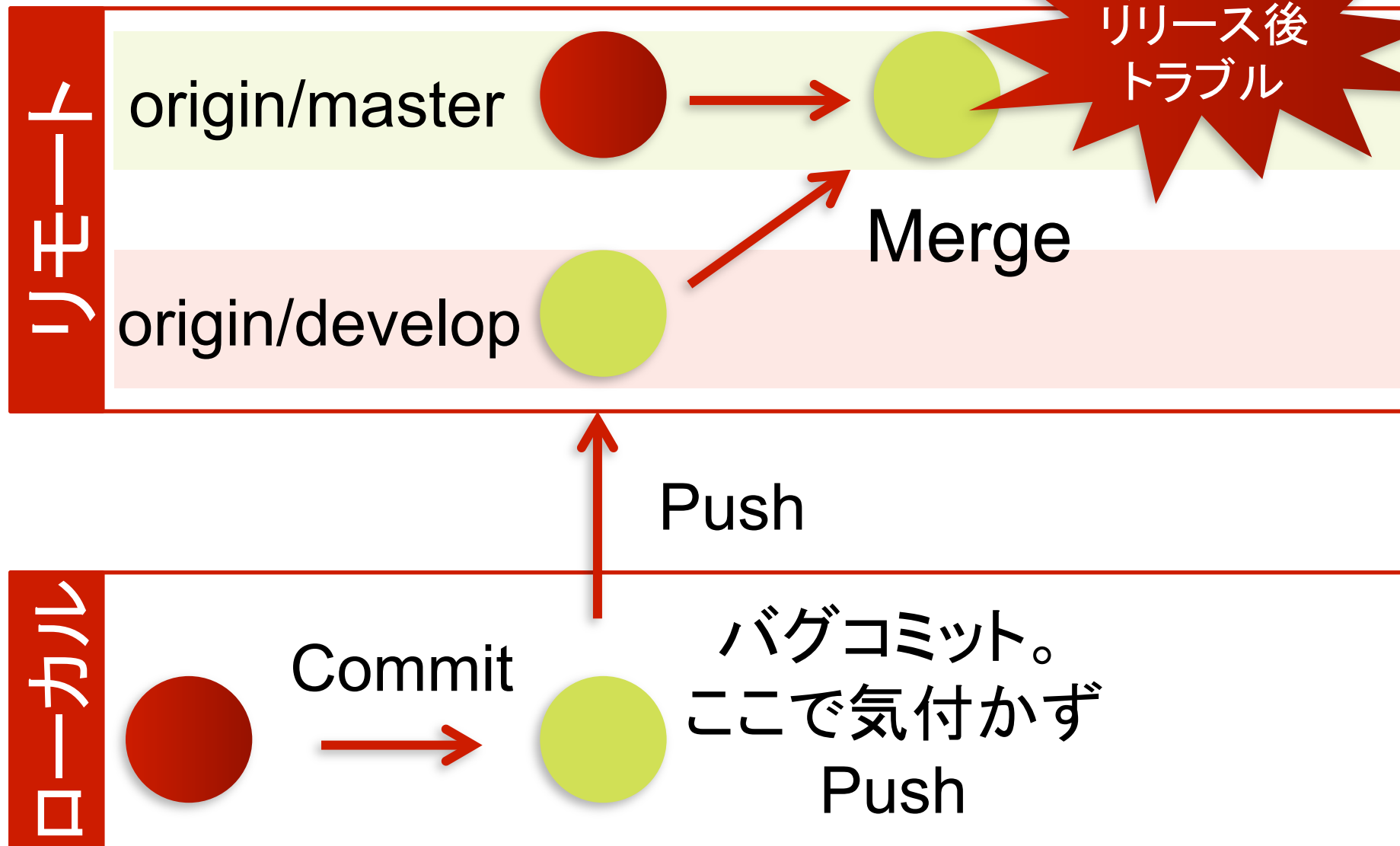
Commit



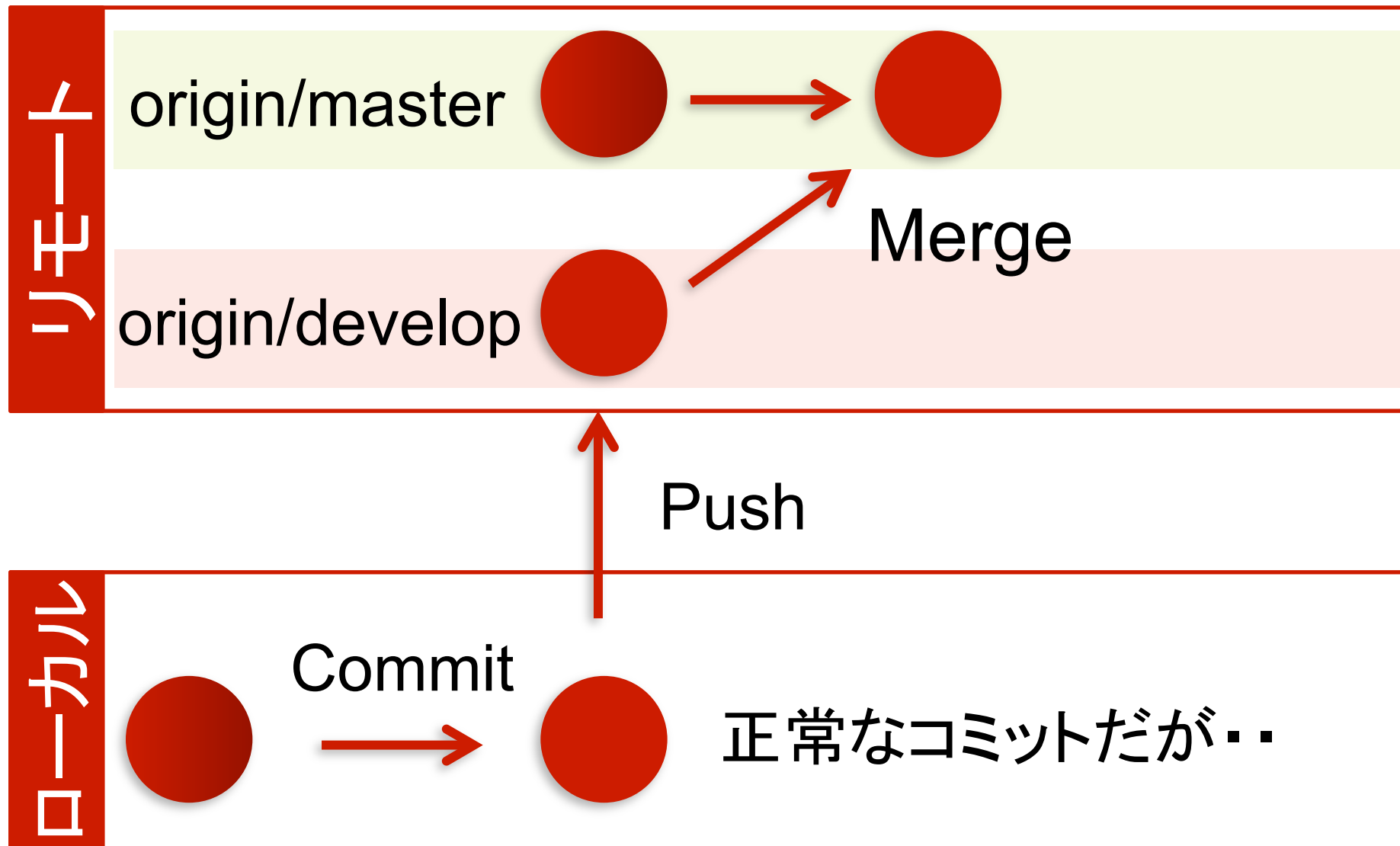
Commit



考えなきゃいけない場合その1



考えなきゃいけない場合その2



考えなきゃいけない場合その2

MGR

「まだリリースしないで！」

リモート

origin/develop

Merge

Push

ローカル

Commit

正常なコミットだが・・・

どう戻すか？

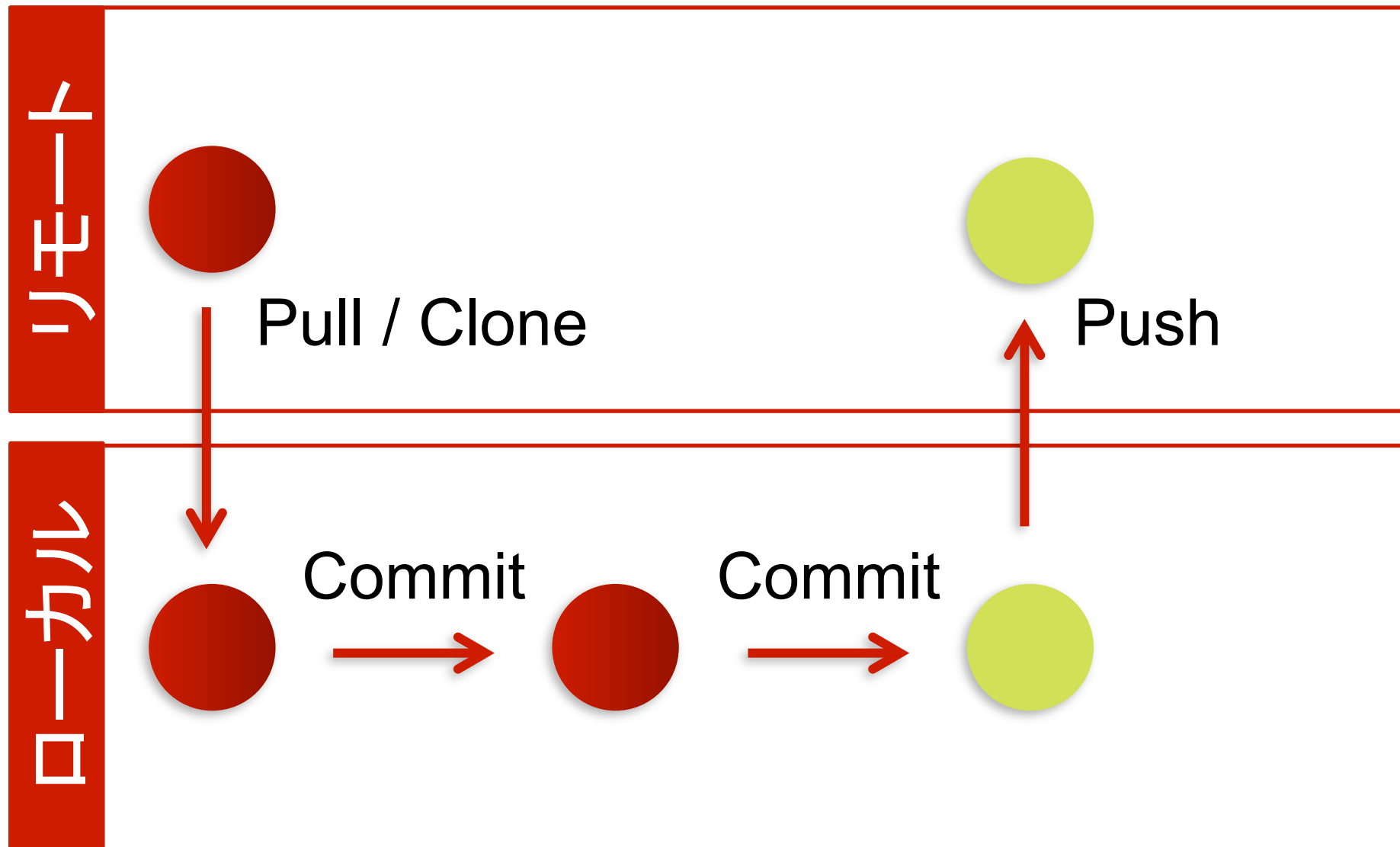
Remoteを戻す方法は2つ

1. 改築してforce push
2. 打消コミットを作ってpush

Remoteを戻す方法は2つ

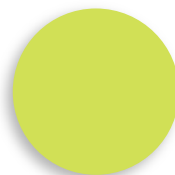
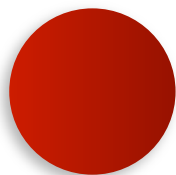
1. 改築してforce push
2. 打消コミットを作ってpush

1. 改築してforce push

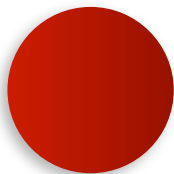
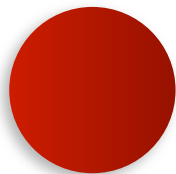


1. 改築してforce push

リモート



ローカル

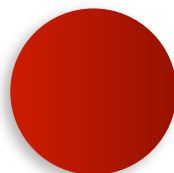
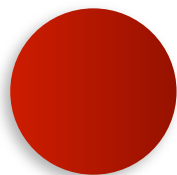


Reset



1. 改築してforce push

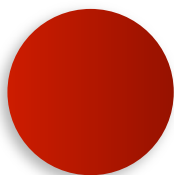
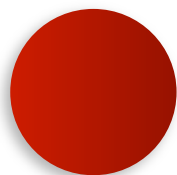
リモート



Gone!

Force push!

ローカル

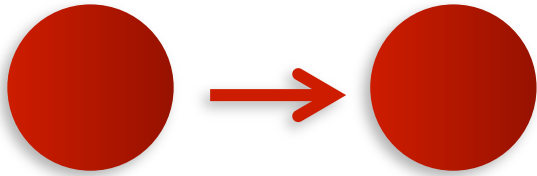


メリット

- コミットグラフが綺麗になる
- 無意味な間違いの記録を抹消できる

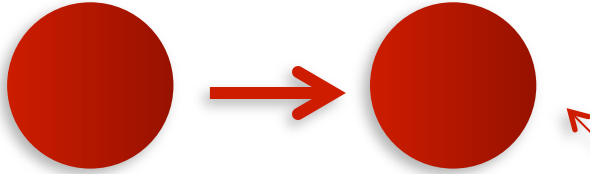
落とし穴

リポート

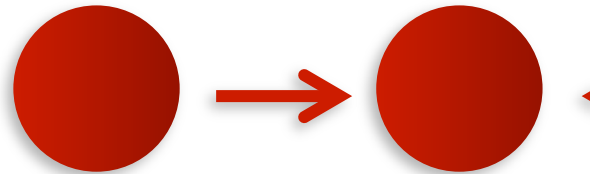


落とし穴

ボブ

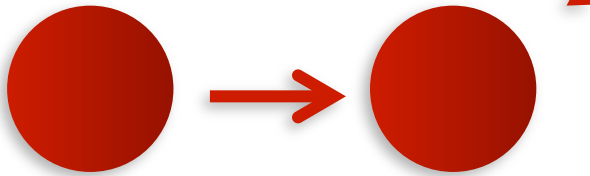


リモート



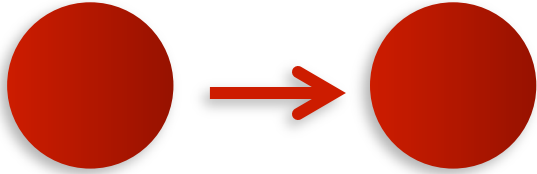
同じコミット

アリス

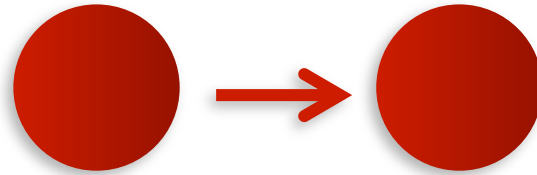


落とし穴

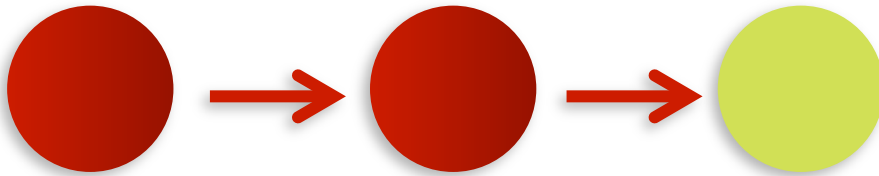
ボブ



リモート

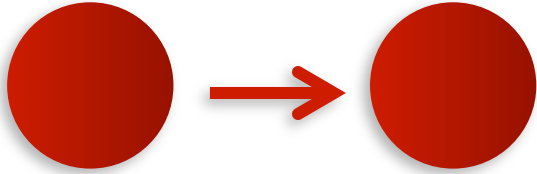


アリス

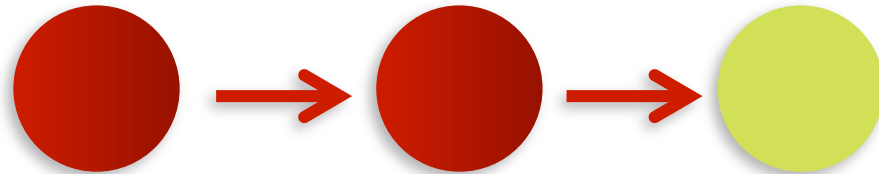


落とし穴

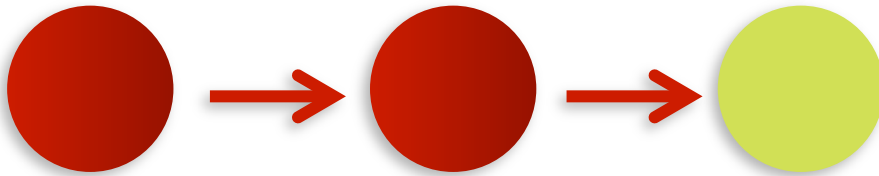
ボブ



リモート



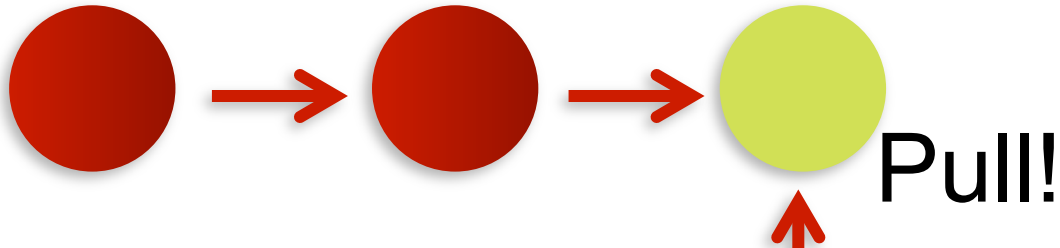
アリス



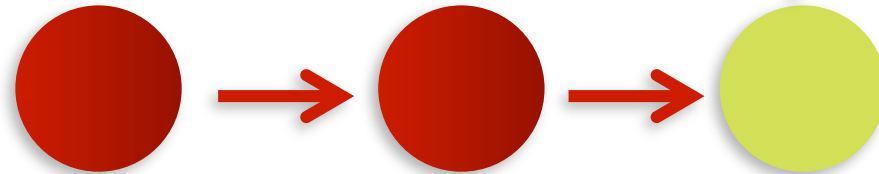
Push!

落とし穴

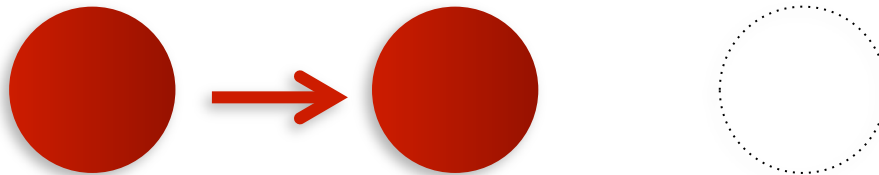
ボブ



リモート

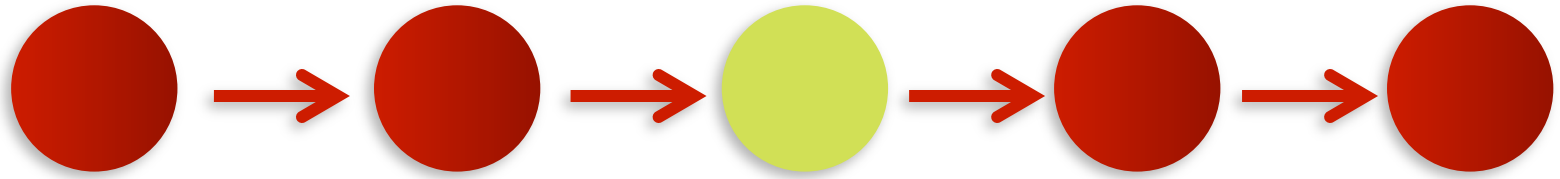


アリス

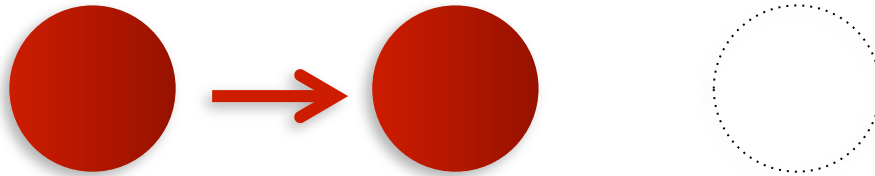


落とし穴

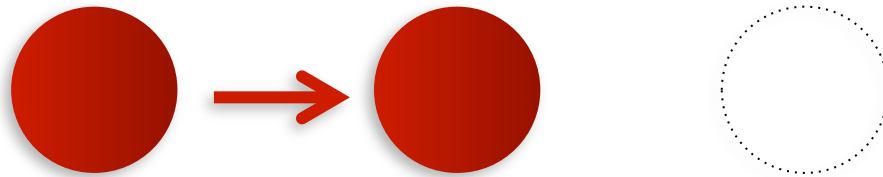
ボブ



リモート



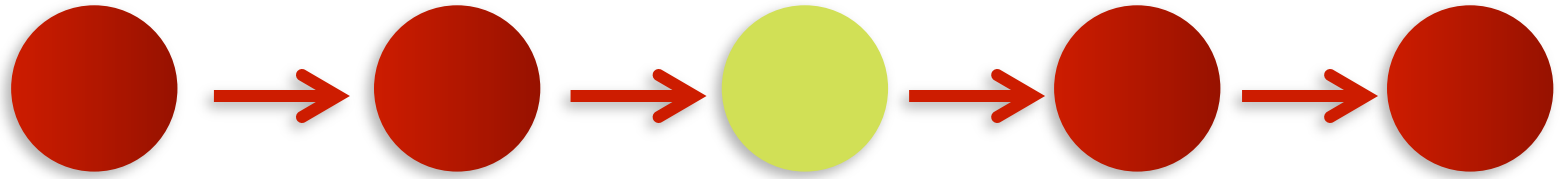
アリス



Force Push!

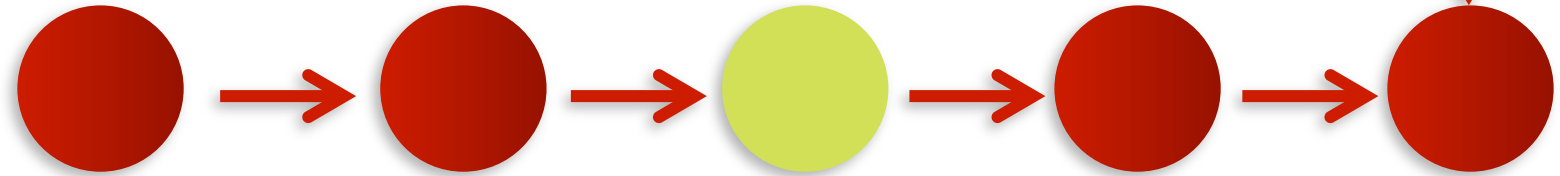
落とし穴

ボブ



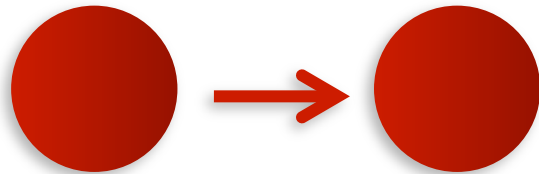
Push!

リモート



Hello!

アリス



怖いところ

- 警告が出ない
- 誰がPullしたか分からない
- いつ復活するか分からない

内緒でやったらほぼ確実に事故る

この方法が適切なケース

＝まだ誰もPullしていない時

- Push(マージ)後すぐに気付いた
- 少人数開発
- すぐに声掛けできる(近くにいる)
- Reset等で簡単に(短時間で)直せる

= 誰かにPullされた後

- Pullした人がreset/rebase/cherry-pick等で直すのはヘビー
 - 事故のもと
 - 全員がgitマスターではない

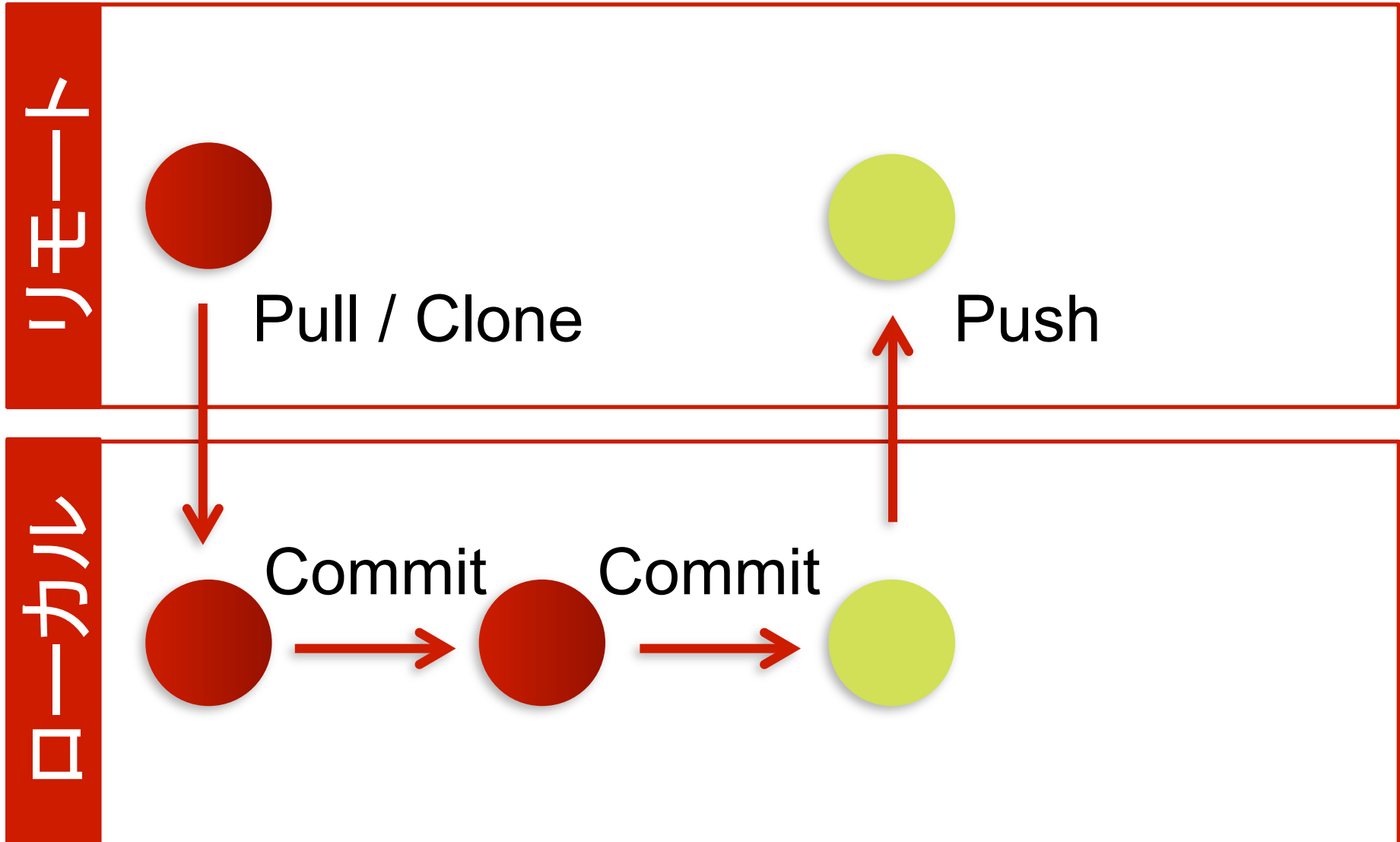
注意点

- 基本リスキー (reset/force-push)
- 簡単にできるときのみ行う
(常用しない)
- 自信が無ければ避けた方が無難

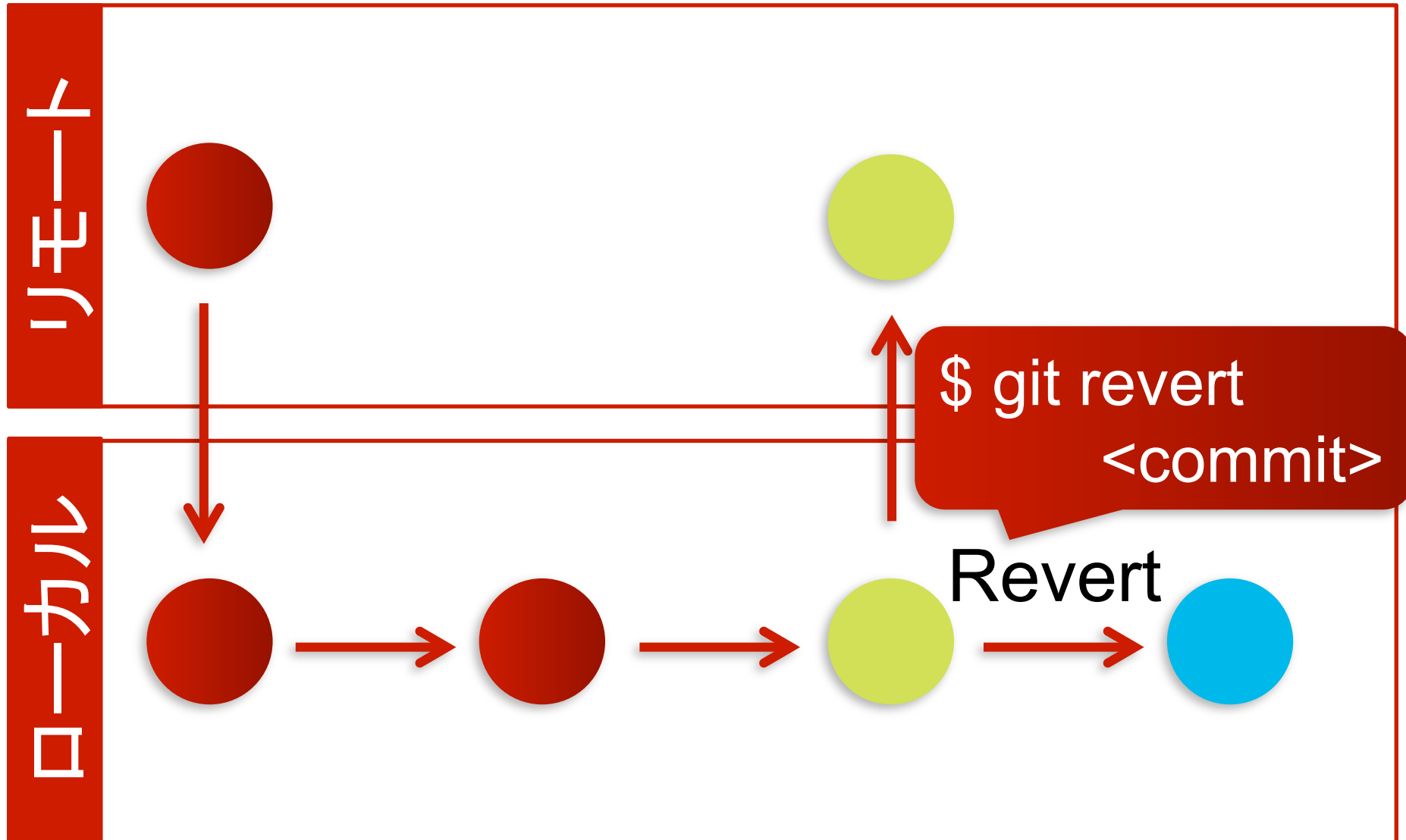
Remoteを戻す方法は2つ

1. 改築してforce push
2. 打消コミットを作ってpush

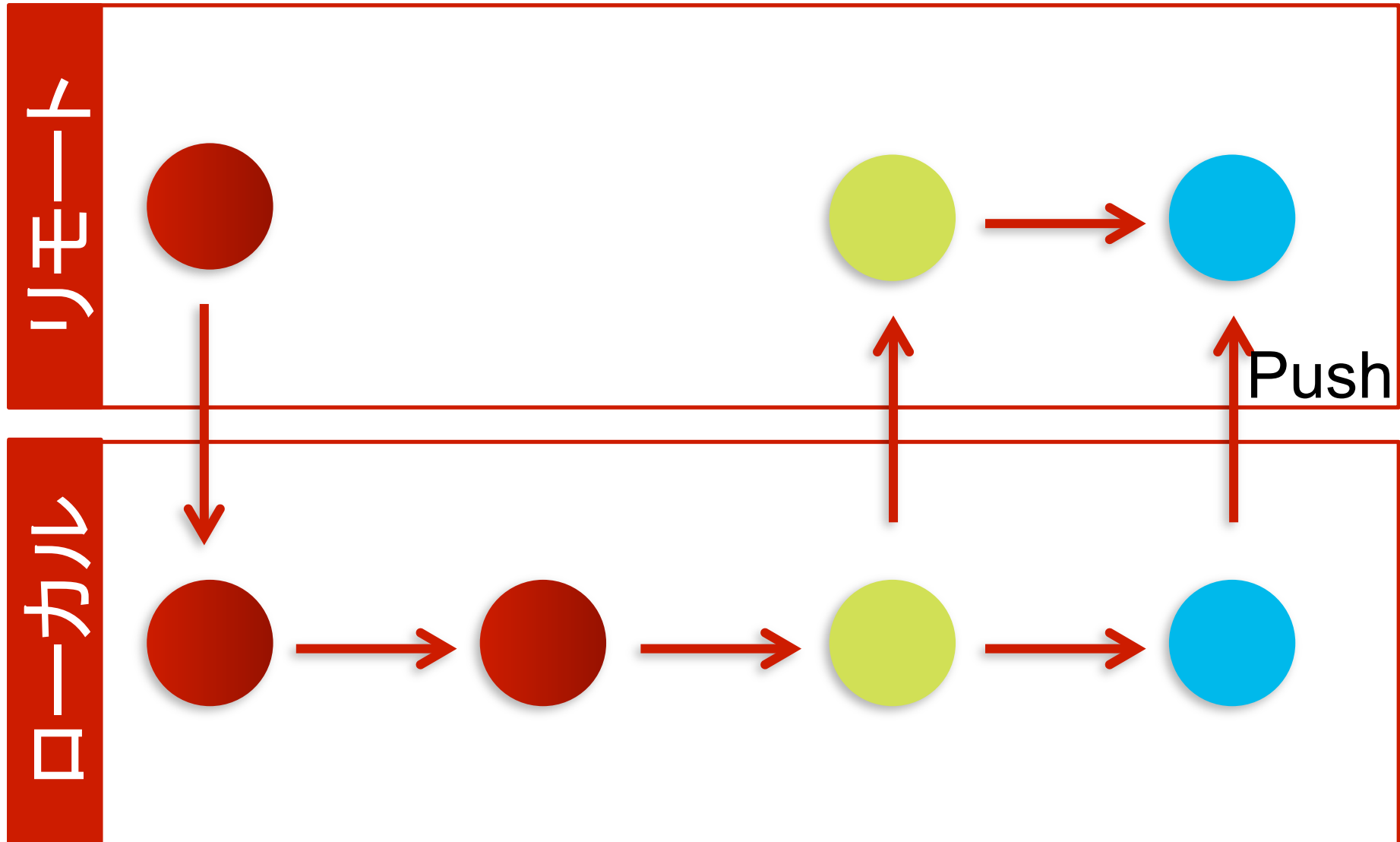
2. 打消コミットを作ってpush



2. 打消コミットを作ってpush



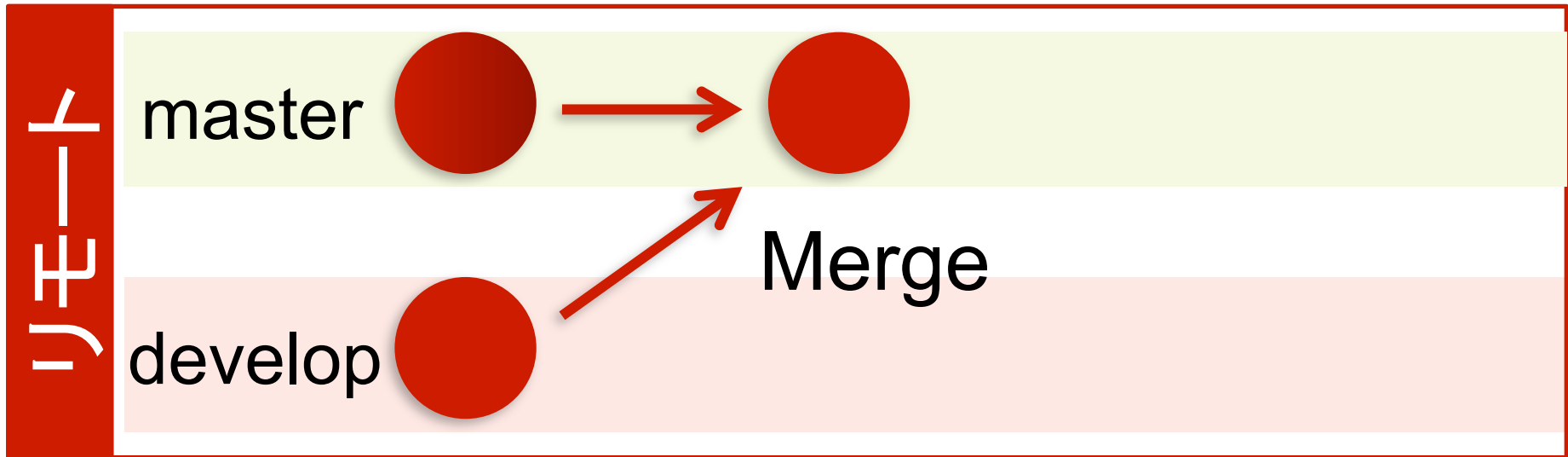
2. 打消コミットを作ってpush



メリット

- 誰かにPullされた後でもOK
 - むしろされた後はこれしかない？
- オペレーションリスクが低い
 - コミットを破壊しなくて済む
 - Force-pushしなくて良い

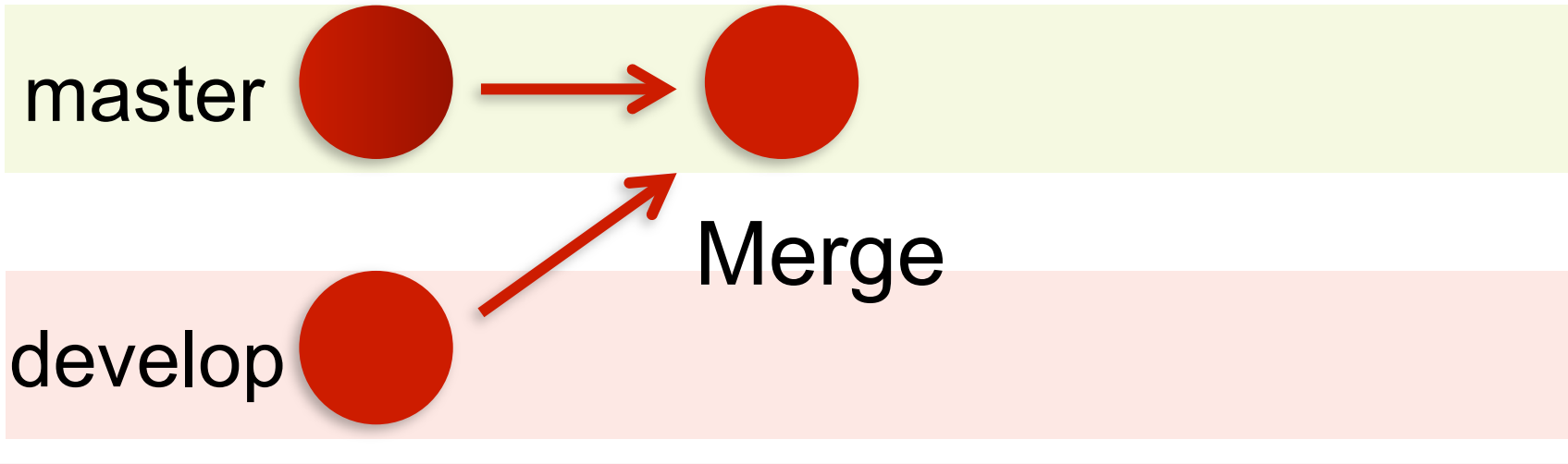
落とし穴その1



MGR

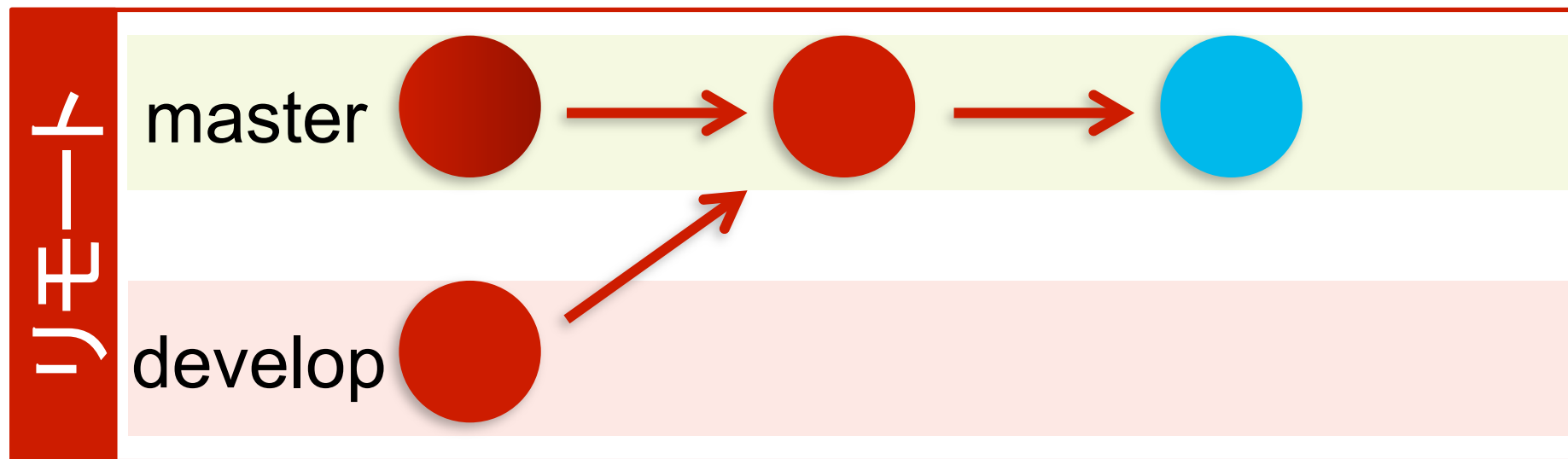
「まだリリースしないで！」

テーマ

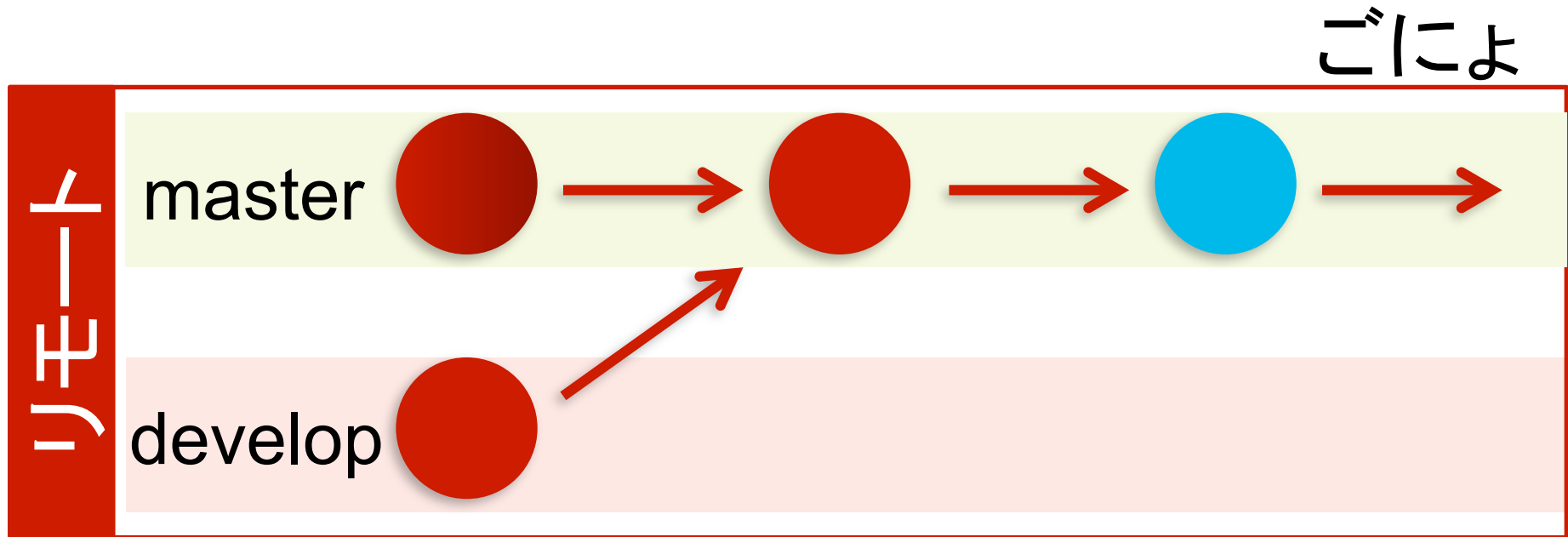


落とし穴その1

Revert



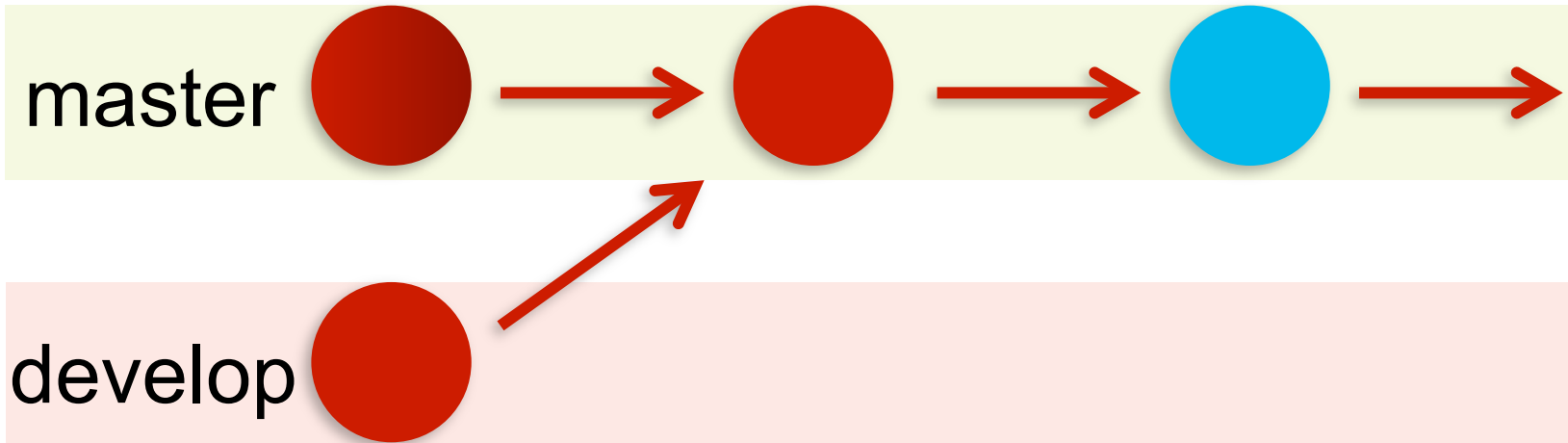
落とし穴その1



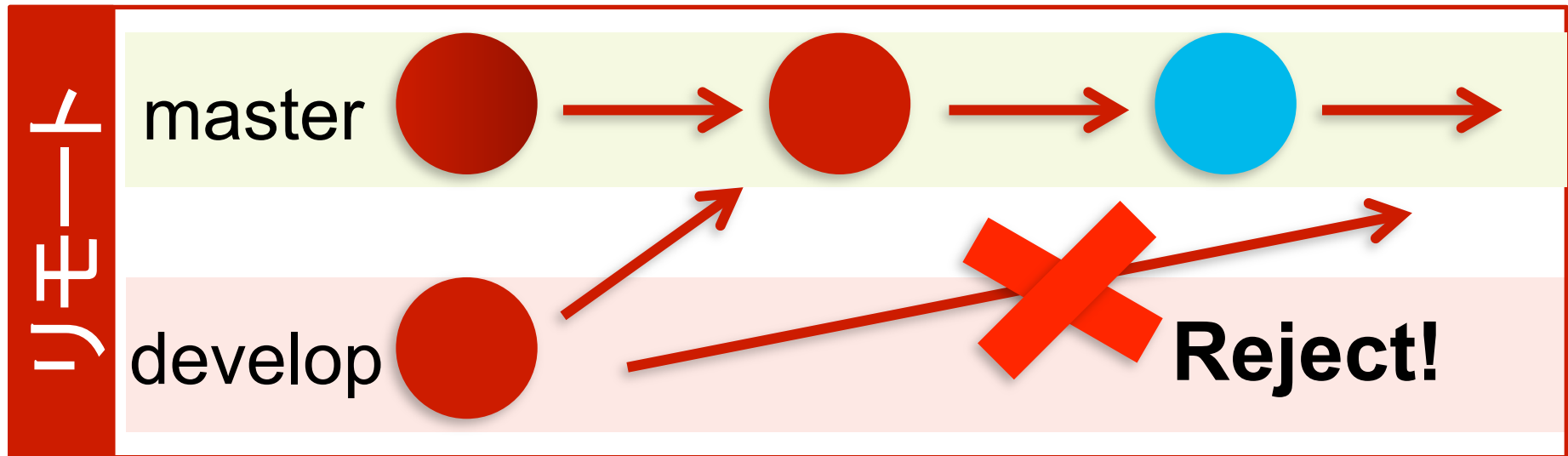
MGR

「いいよ！」

フロー



既にマージされているので
リジェクトされる



怖いところ

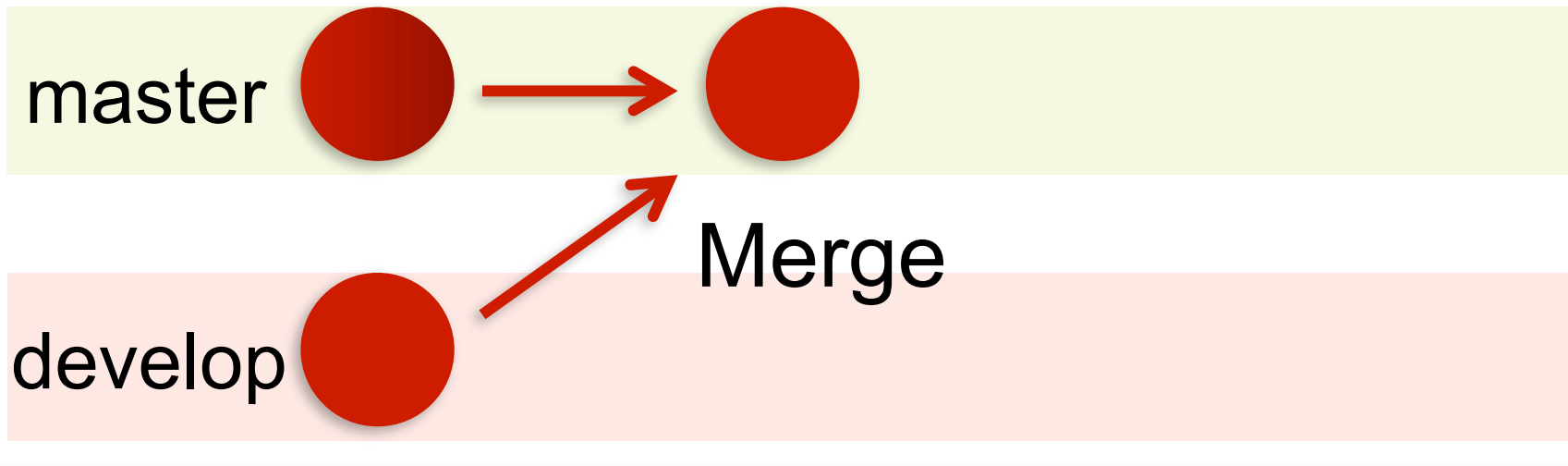
- 戻した後に戻せない

こうしましょうその1

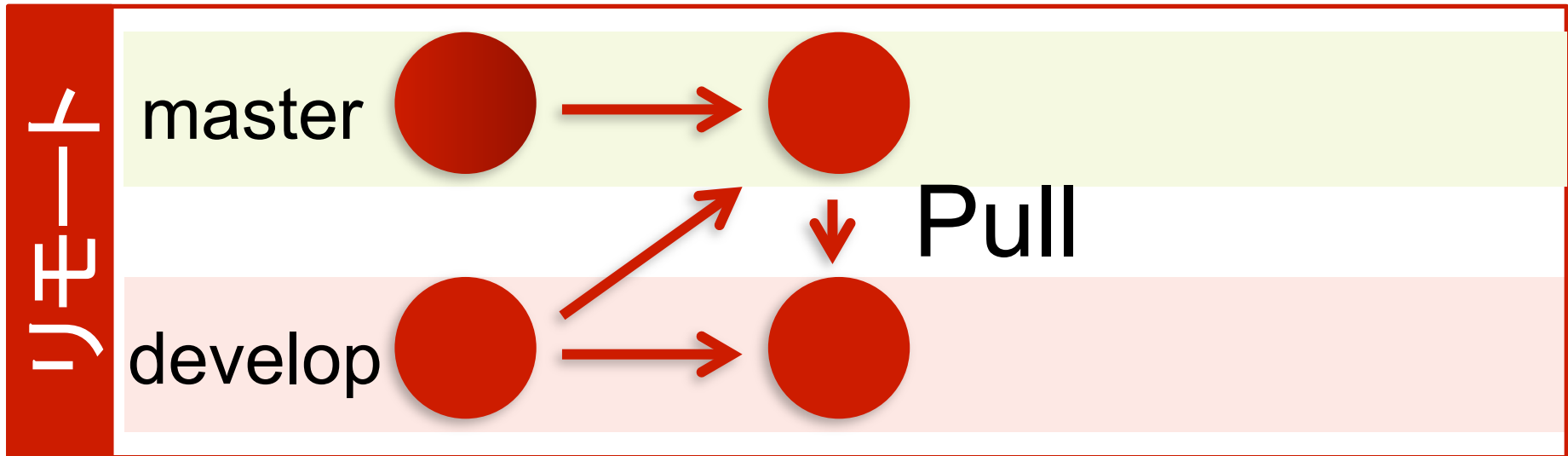
MGR

「まだリリースしないで！」

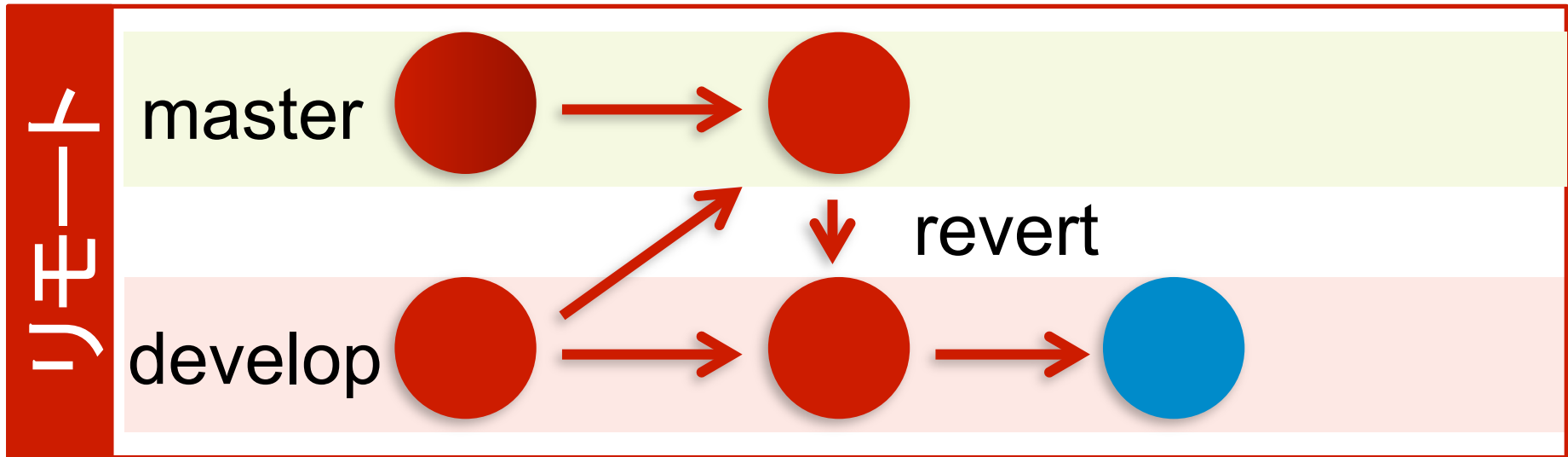
テーマ



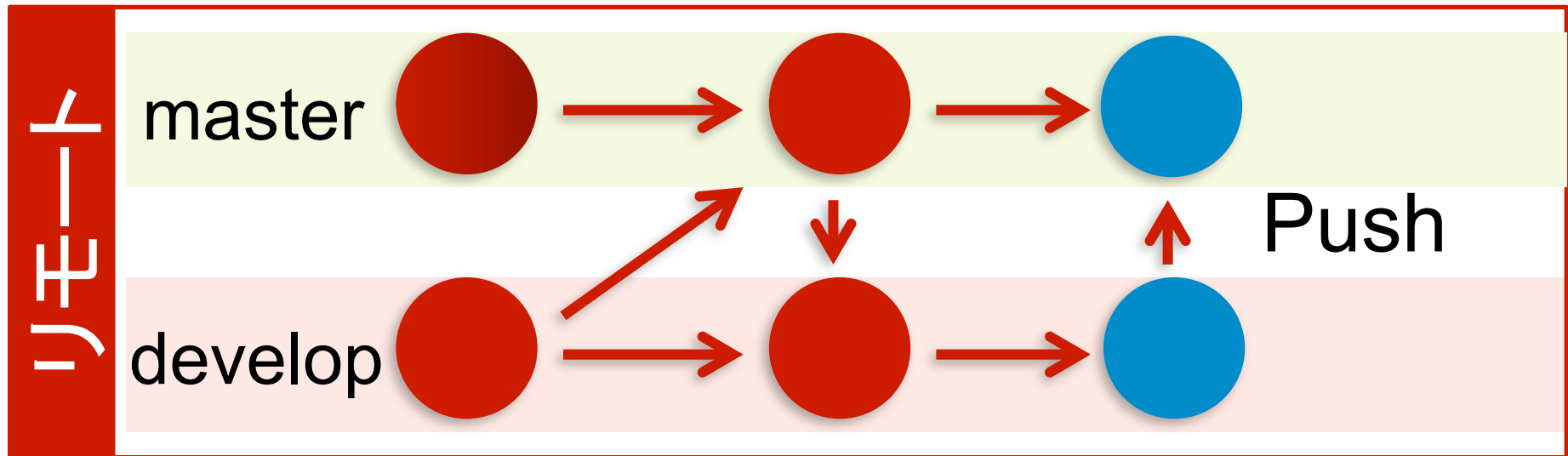
こうしようその1



こうしましょうその1



こうしましょうその1

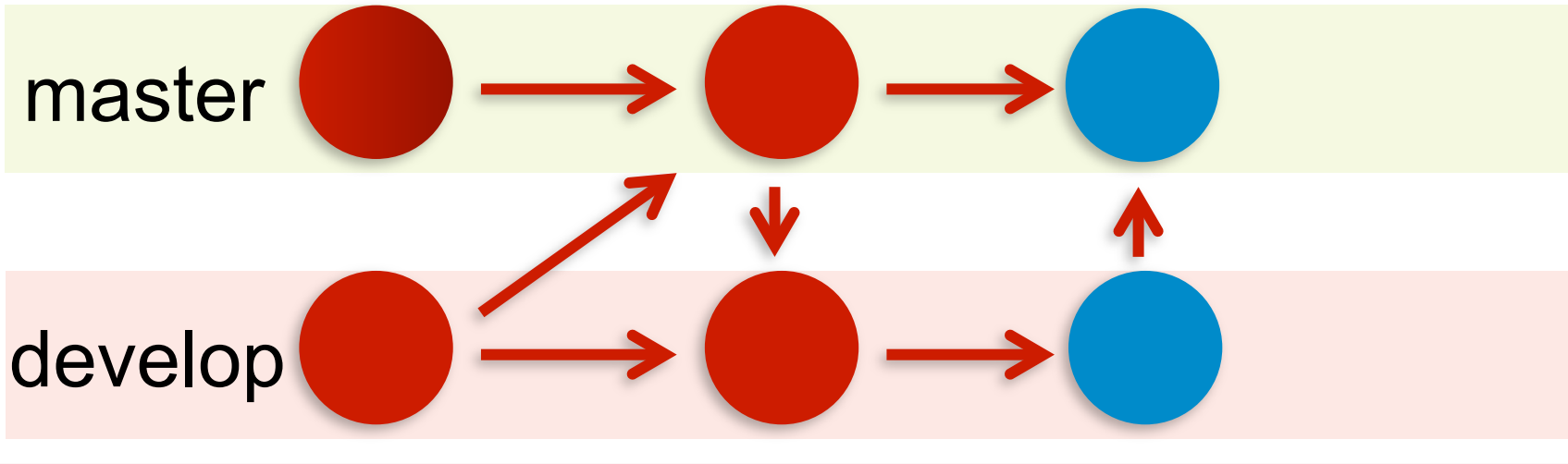


こうしましょうその1

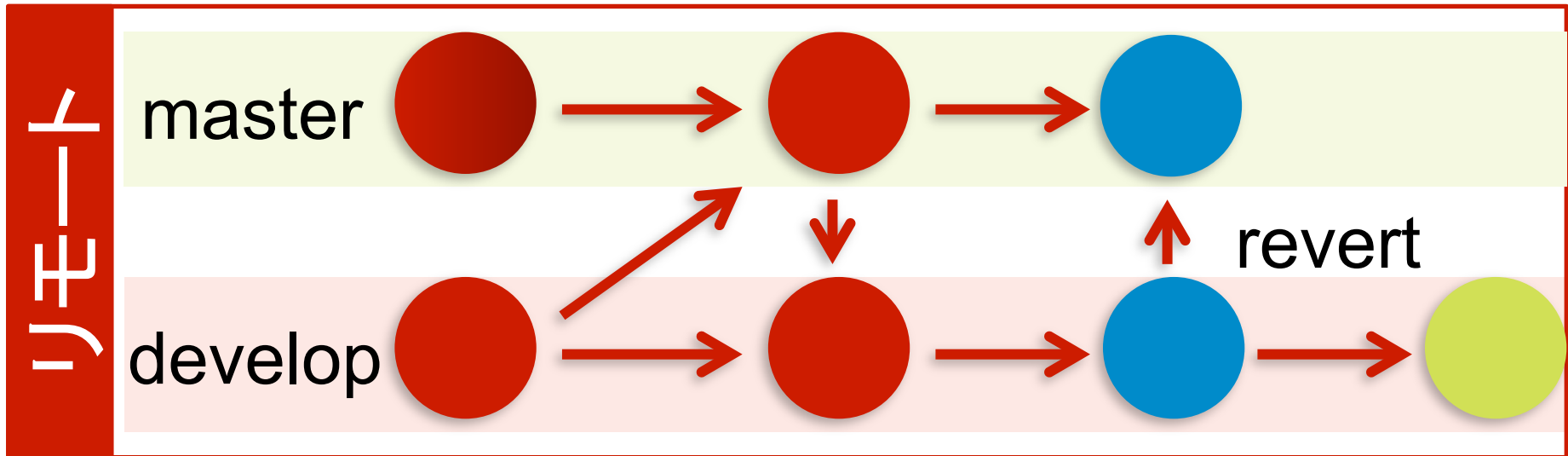
MGR

「いいよ！」

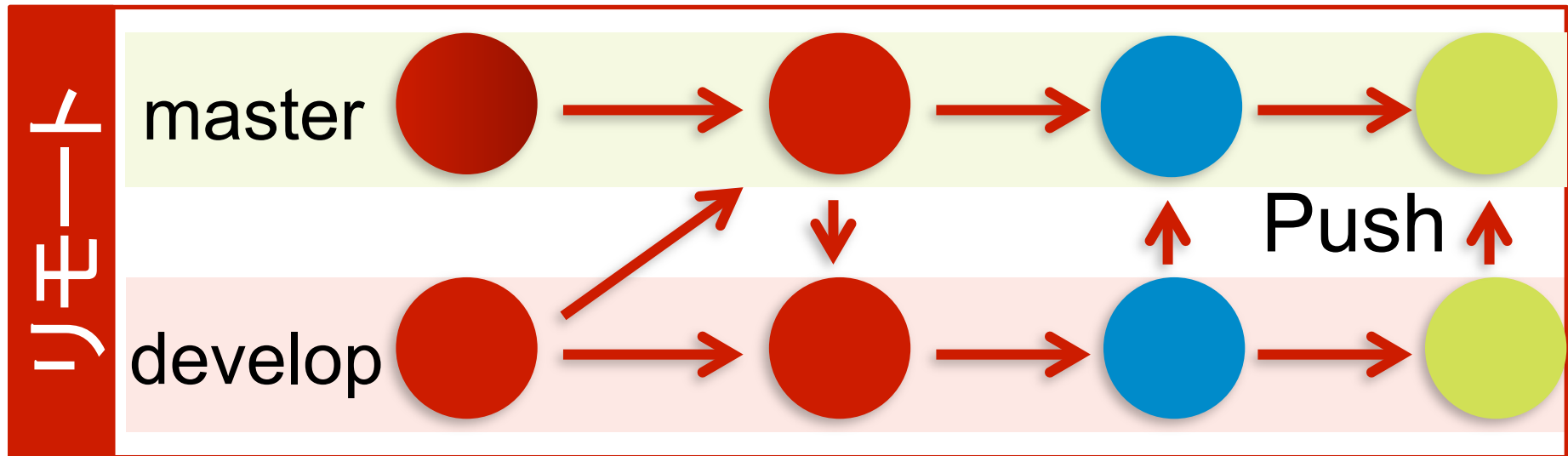
トータル



こうしましょうその1



こうしようその1

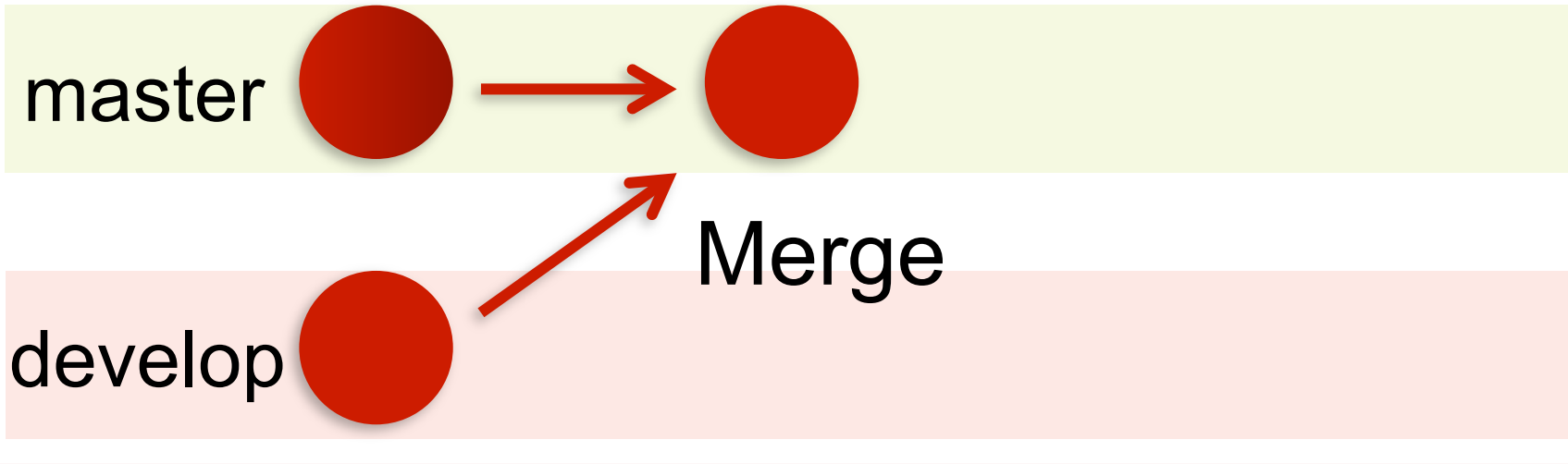


- revertのrevertができる
- コミットグラフは汚くなる

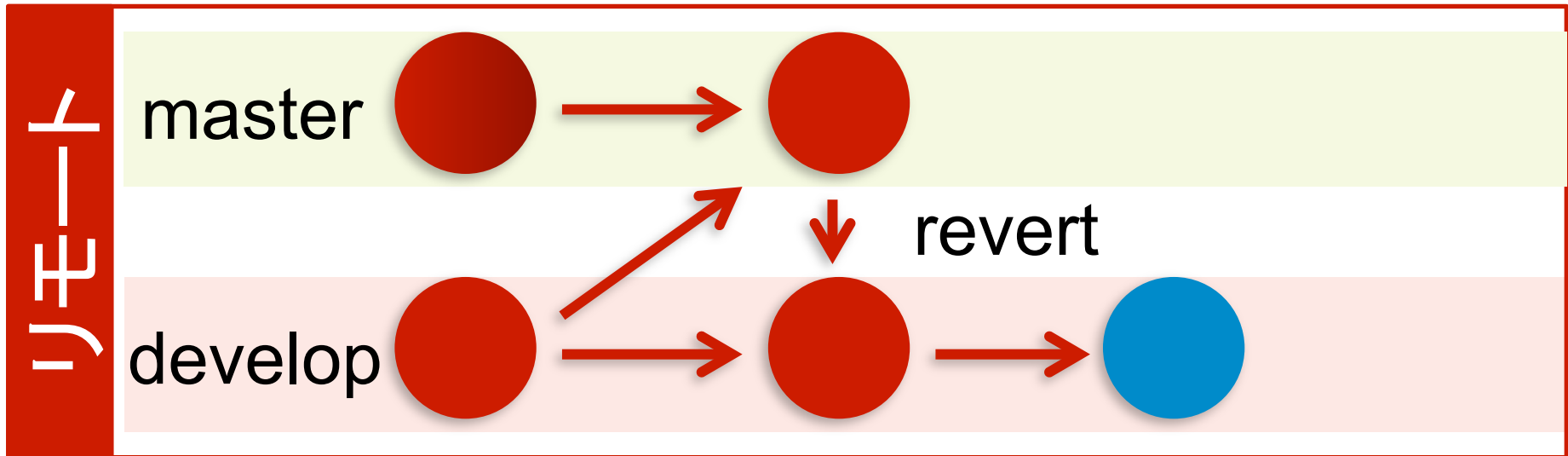
MGR

「まだリリースしないで！」

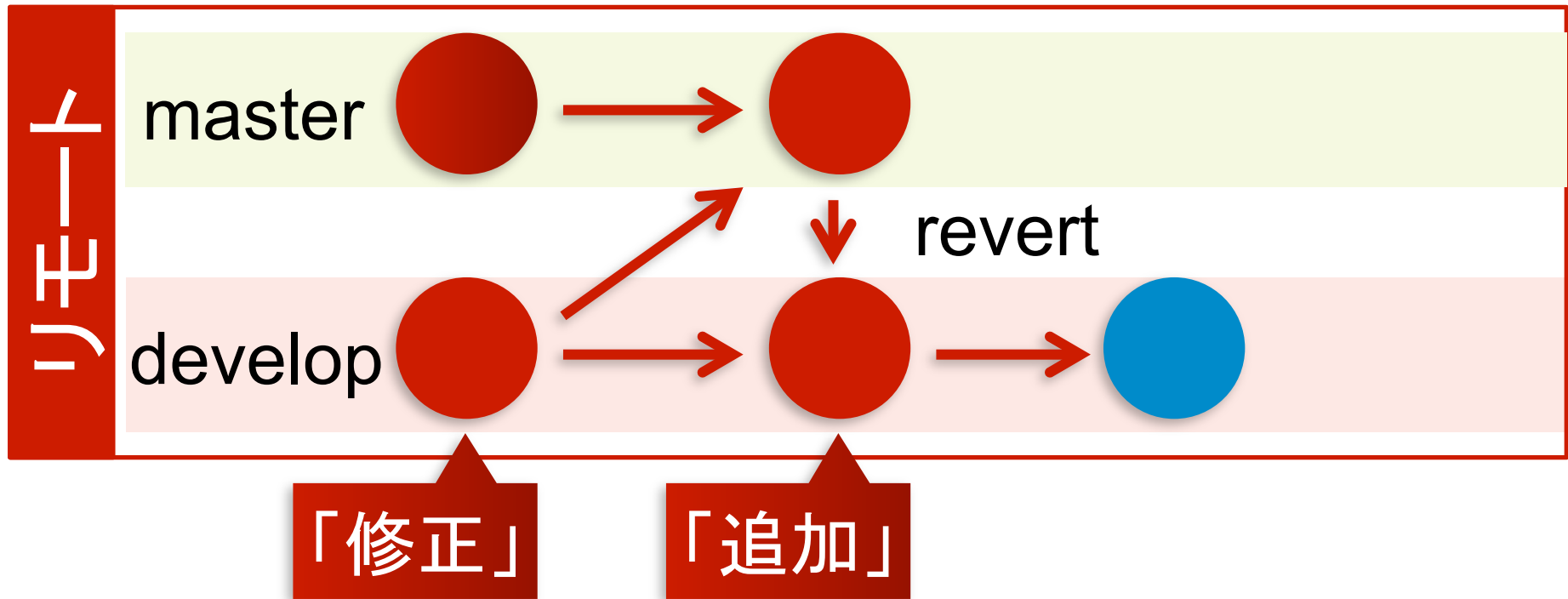
テーマ



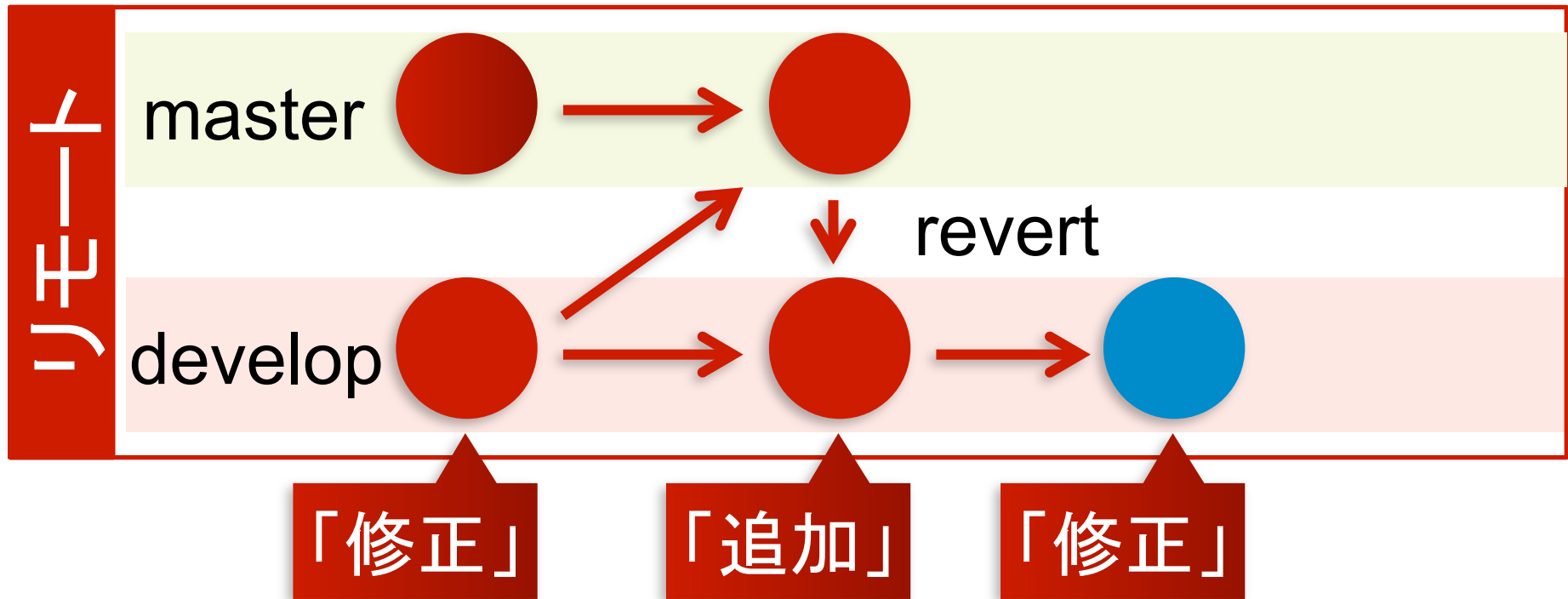
落とし穴その2



落とし穴その2

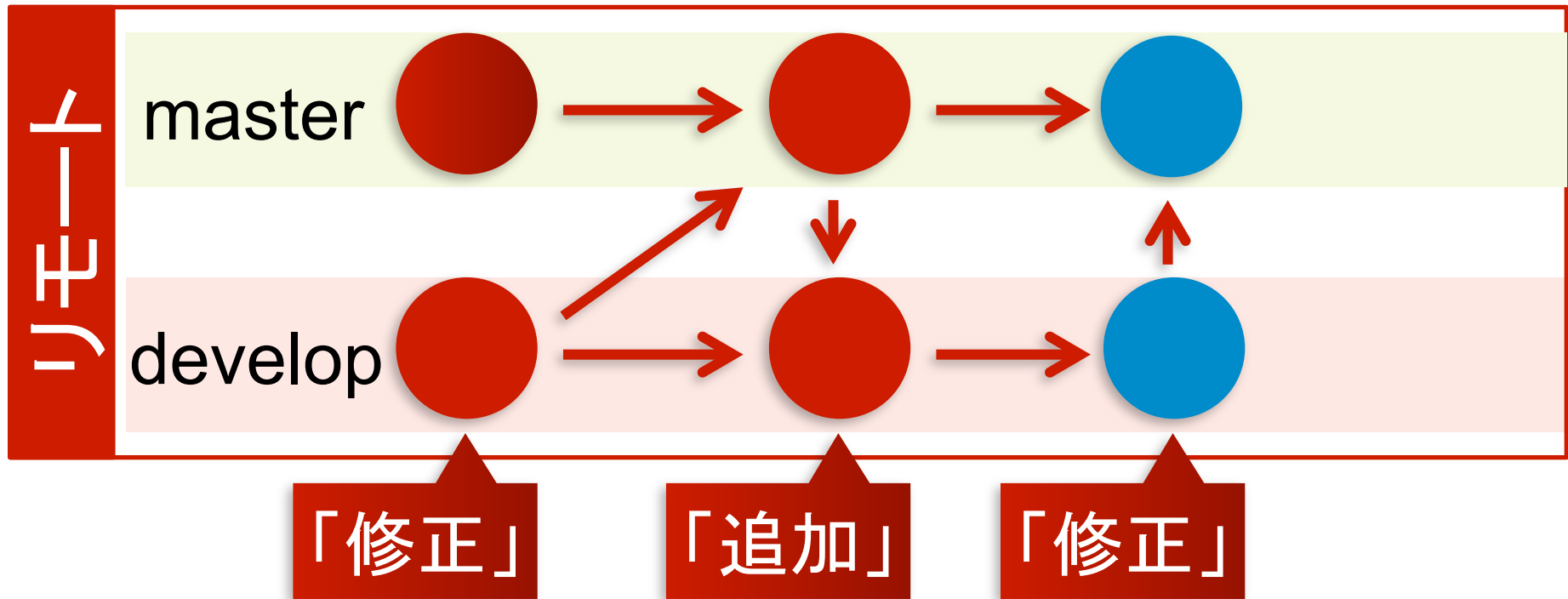


落とし穴その2



MGR

「いいよ！」

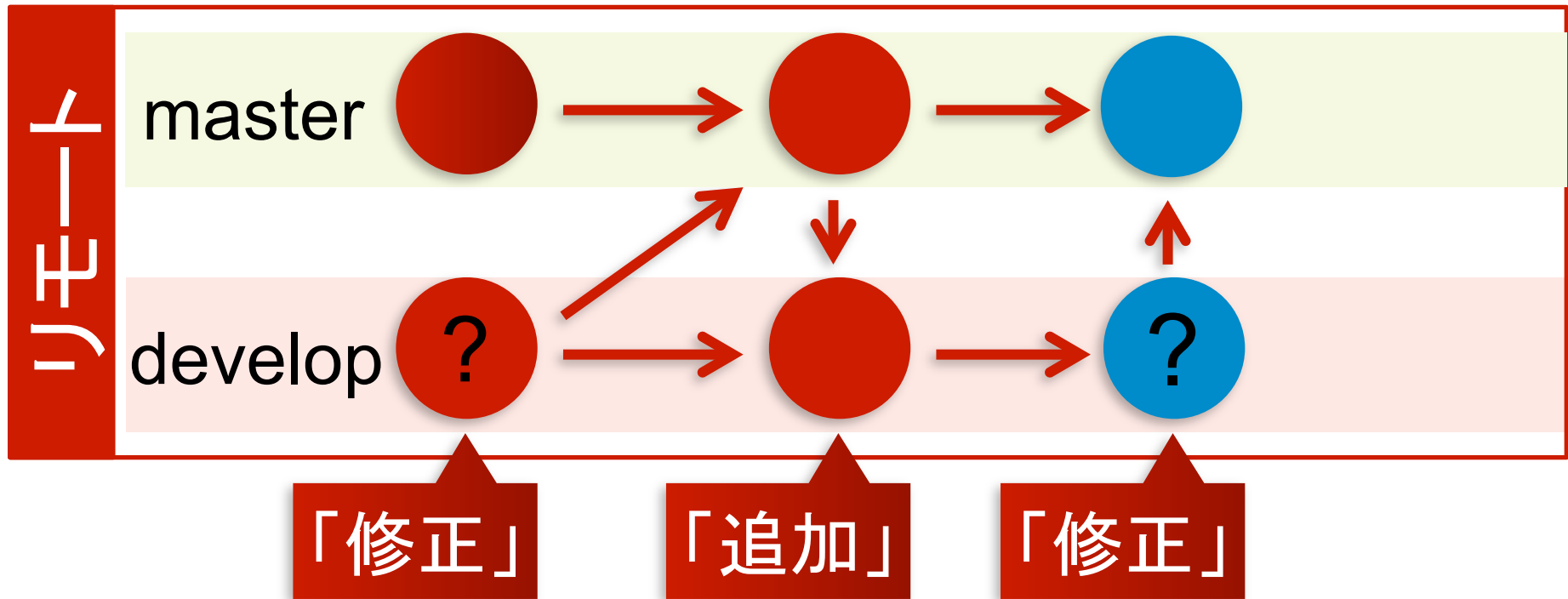


落とし穴その2

MGR

「いいよ！」

どれだっけ？



怖いところ

- コミットメッセージを適当に編集すると普通のコミットと見分けがつかない
- その上時間が経つと...

この方法が適切なケース

＝Force-pushが使え(わ)ない場合

- 間違いコミットをPullされた
- メンバーをすぐ集めるのが困難
- masterが相手

この方法がダメなケース

=ルール化していない場合

- コミットメッセージを書き換えたり。
(基本デフォルトのままで...)
- 勝手にmasterのコミットをrevertしたり。
(基本masterに直コミットは禁止。)

まとめ

- 声をかけてさくっとできそうならforce-push (ただしリスクは考える事)
- それ以外はrevertだがルールを整備しないと戻せなくなる(コミットメッセージ, etc...)

何も話さず、勝手にやらないこと！

おわりに

この度貴重な発表の場を用意して下さいましたDevLOVE現場甲子園2014実行委員の皆様、
並びに会場を提供して下さいました株式会社Oracle様に厚くお礼申し上げます。

ご清聴誠にありがとうございました

Thank you very much for your kind attention!