

GitとSubversionの認証認可

H3 Partners, Inc.
たにやん

2026.8.29

はじめに

1. OSSplaza.comでは、Git, Subversionの技術サポートを行っている
2. 頂く質問のほとんどはサーバーの管理関連で、「バージョン管理の機能」について質問されることはまずない
3. 本日はよく「教えて」と言われる、テーマの中から認証認可についてお話してみようと思います。

0. Table of Contents

1. Git, Subversionについて
2. 認証と認可
3. Subversionの認証認可
4. Gitの認証認可
5. Git3.0について少し(認証認可と関係ないですが・・・)
6. BlueOptima ソリューション

1. Git, Subversionについて

1.1 Gitについて

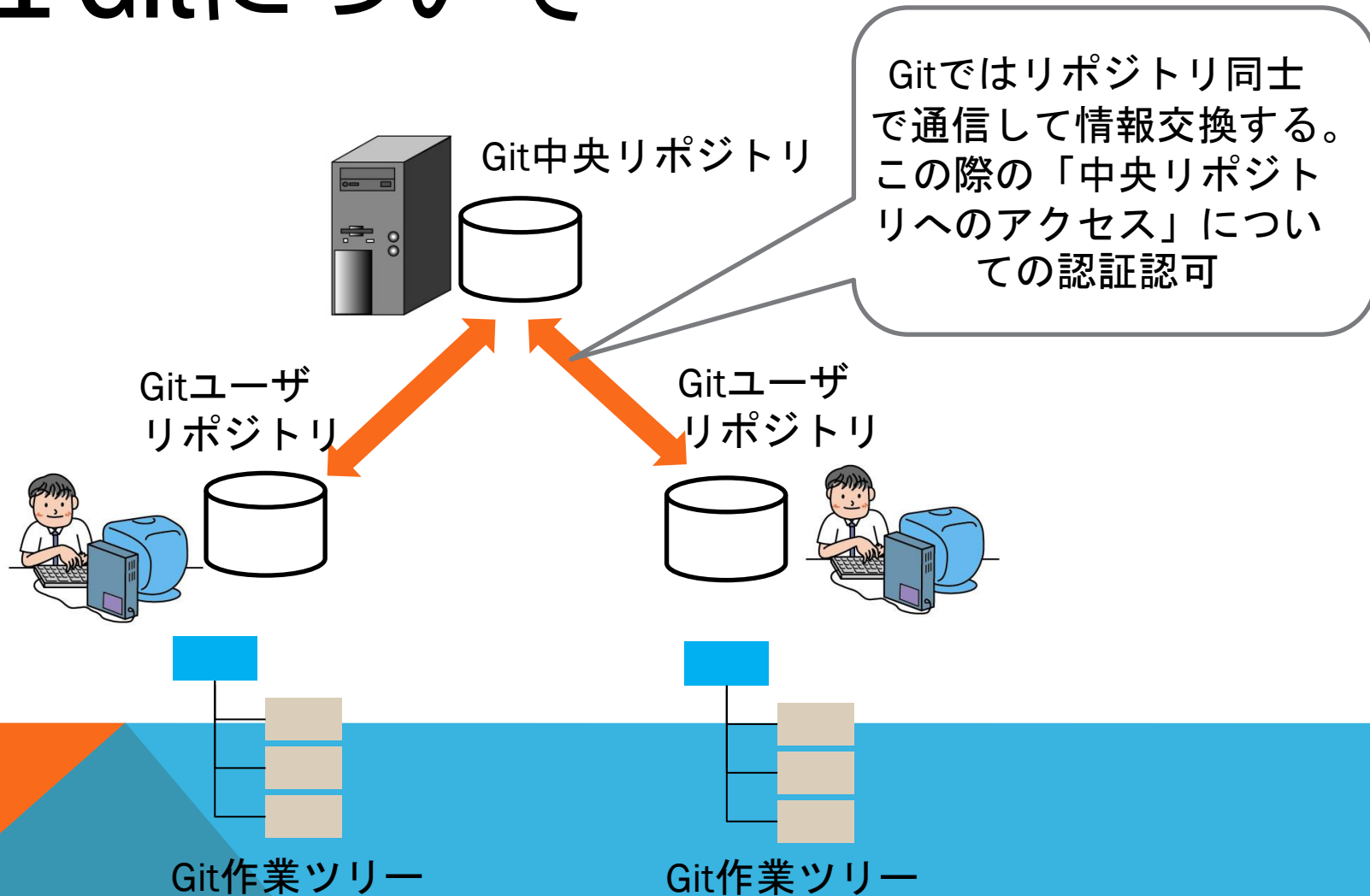
Gitは、近年よく使われているバージョン管理ソフトウェアです。

「**分散型**」と呼ばれるように、個々のプログラム開発者のPCにもリポジトリを置いて、中央リポジトリという共通のリポジトリと更新情報を交換します。

Gitについては「中央リポジトリ」の認証認可を考慮します。

1. Git, Subversionについて

1.1 Gitについて



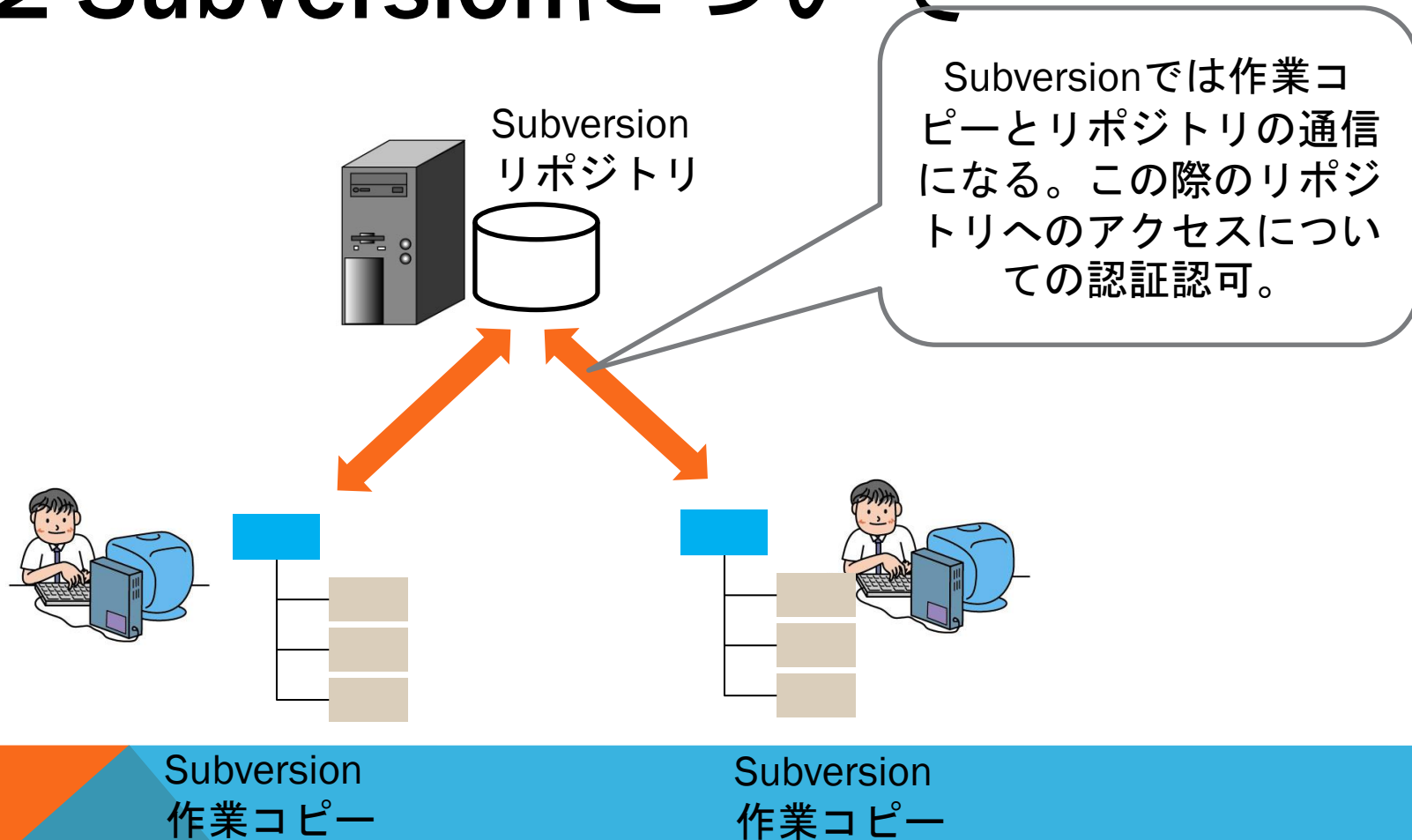
1. Git, Subversionについて

1.2 Subversionについて

- SubversionはGitの前に登場したバージョン管理ソフトウェアで、新規導入は少なくなりましたが現在でも多くのサーバが稼働しています。
- 「**集中型**」と呼ばれていて、各ユーザは「作業コピー」だけを持ってリポジトリはGitという中央リポジトリだけの構成になります。

1. Git, Subversionについて

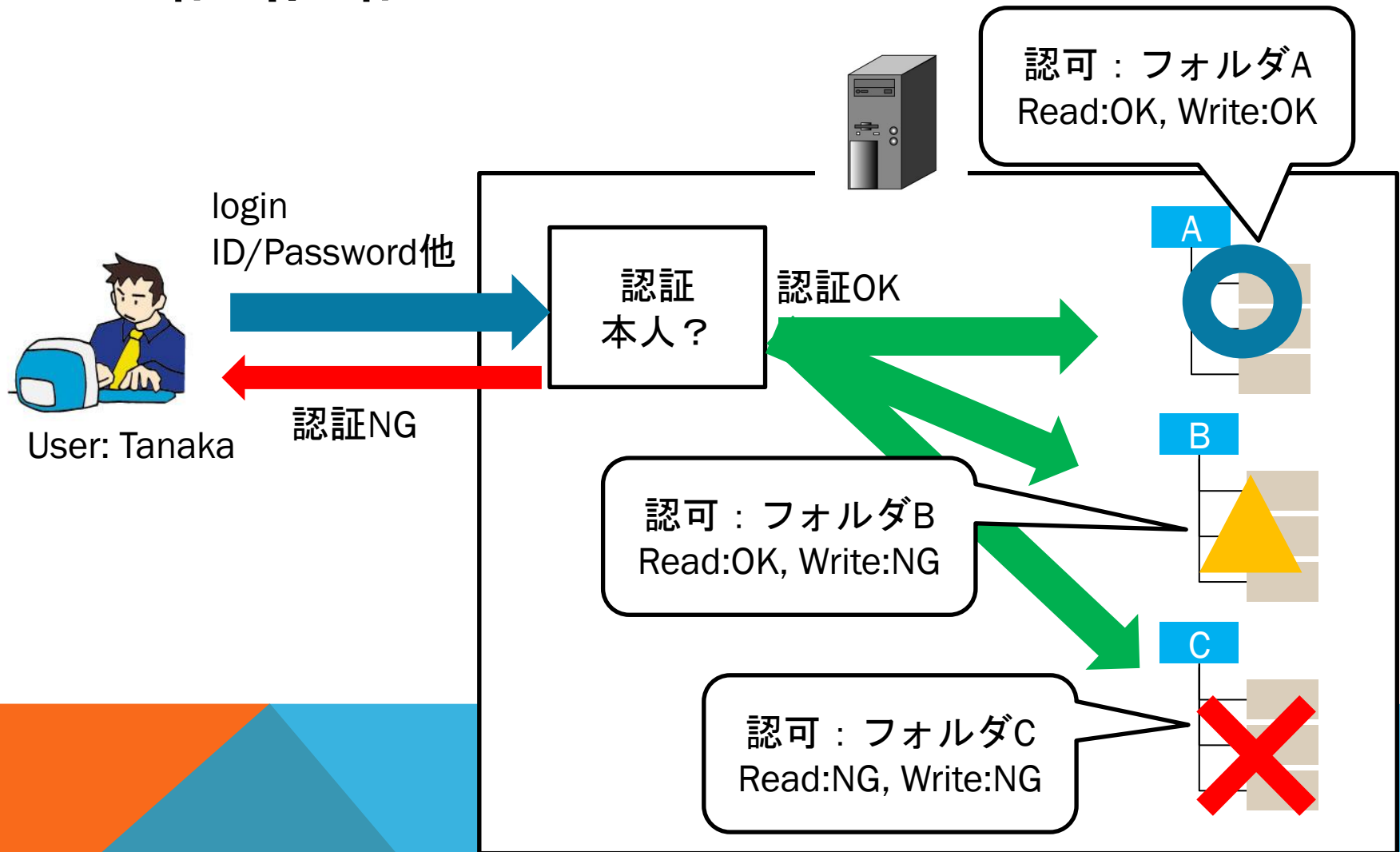
1.2 Subversionについて



2. 認証認可

- **認証** (Authentication: AuthN)
「まちがいなく本人であることを確認」すること
- **認可** (Authorization: AuthZ)
「その人が何にアクセスできるか」を制御すること（アクセス権制御とかいう）
- ただ、実際の設定は入り混じる

2. 認証認可



3. Subversionの認証認可

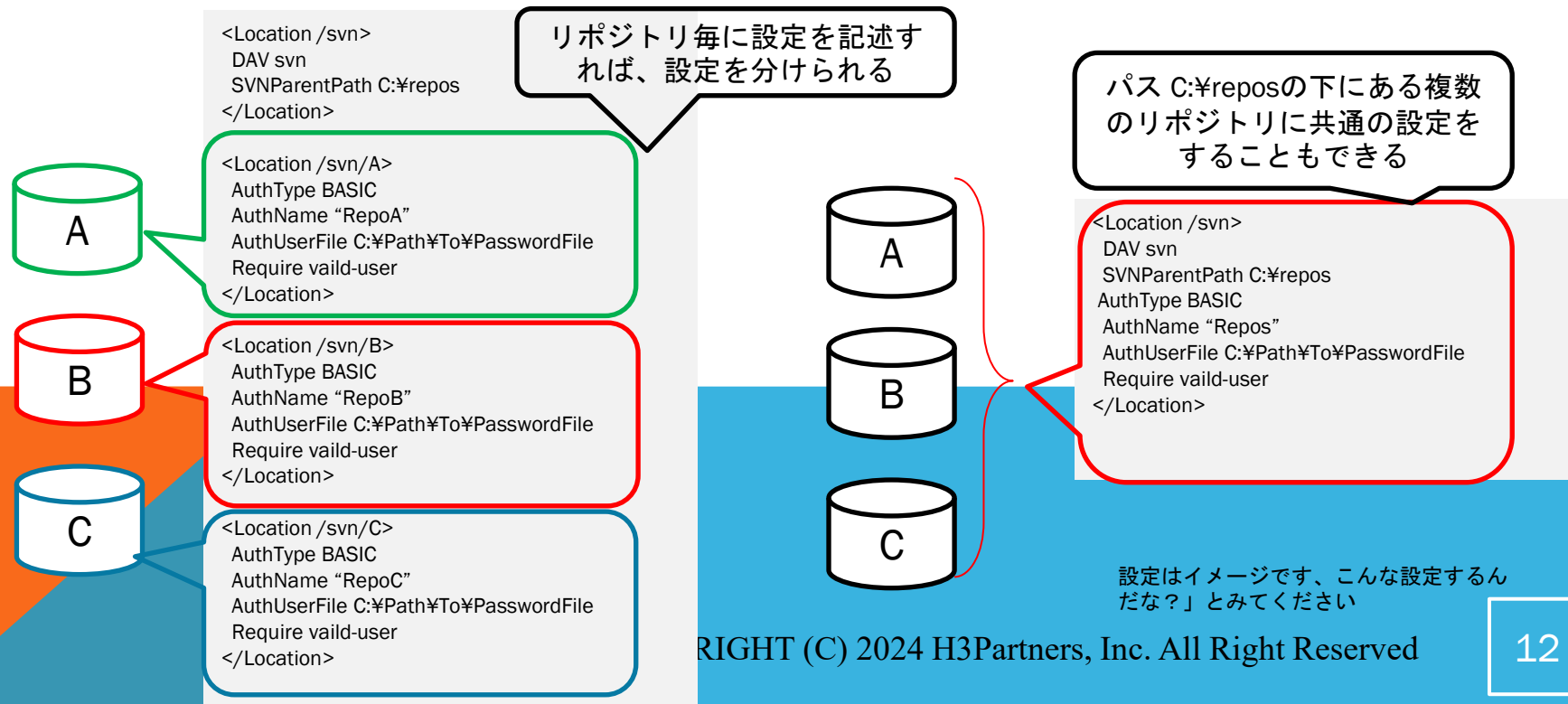
3.1 Subversionの認証

- Subversionサーバに接続する際には、Http(s)かsvnserveのプロトコルを選択します。ほとんどの場合http(s)なのでhttp(s)について説明します。
- http(s)で接続する場合、Apacheの認証に依存します。
- 大きく分けて、
- Apacheベーシック（基本）認証
- LDAP認証
- Windowsドメイン認証などがあります。

3. Subversionの認証認可

3.1 Subversionの認証

- Apacheの設定ファイル(httpd.confとかsubversion.confなど)に認証設定を記載する。
- その際、「リポジトリ群にまとめて設定」と「リポジトリ毎に設定」の方法がある。‘

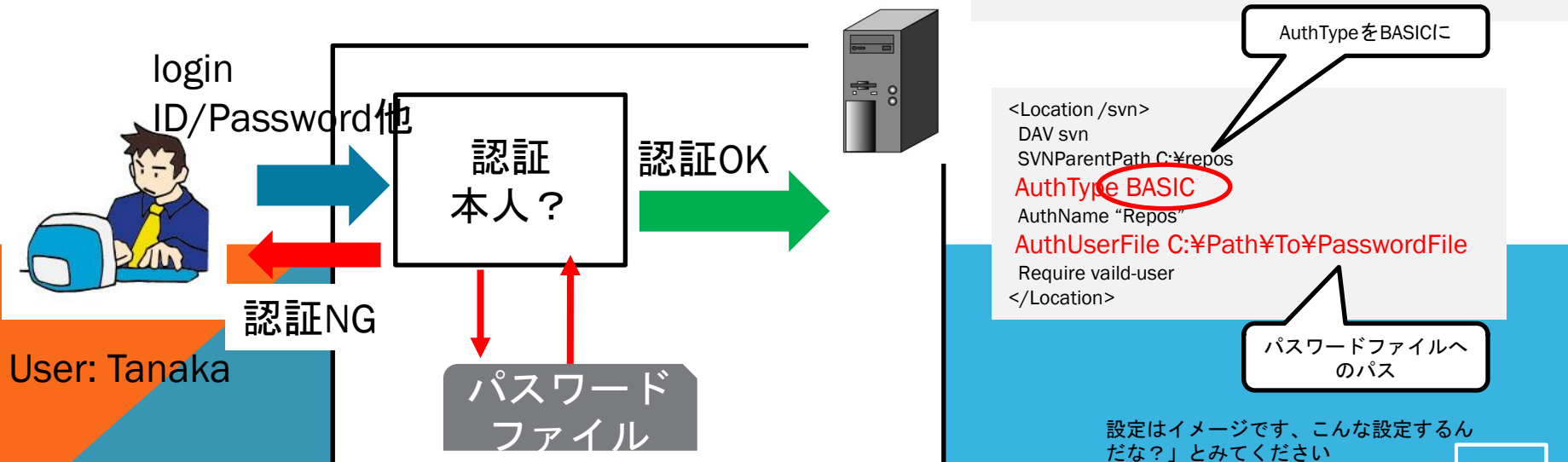


3. Subversionの認証認可

3.1 Subversionの認証

- Apacheベーシック（基本）認証

Apacheの機能で、本人確認のためにパスワードファイルを参照する。
パスワードファイルにはユーザIDやパスワードが確認されており、
それと照合する。

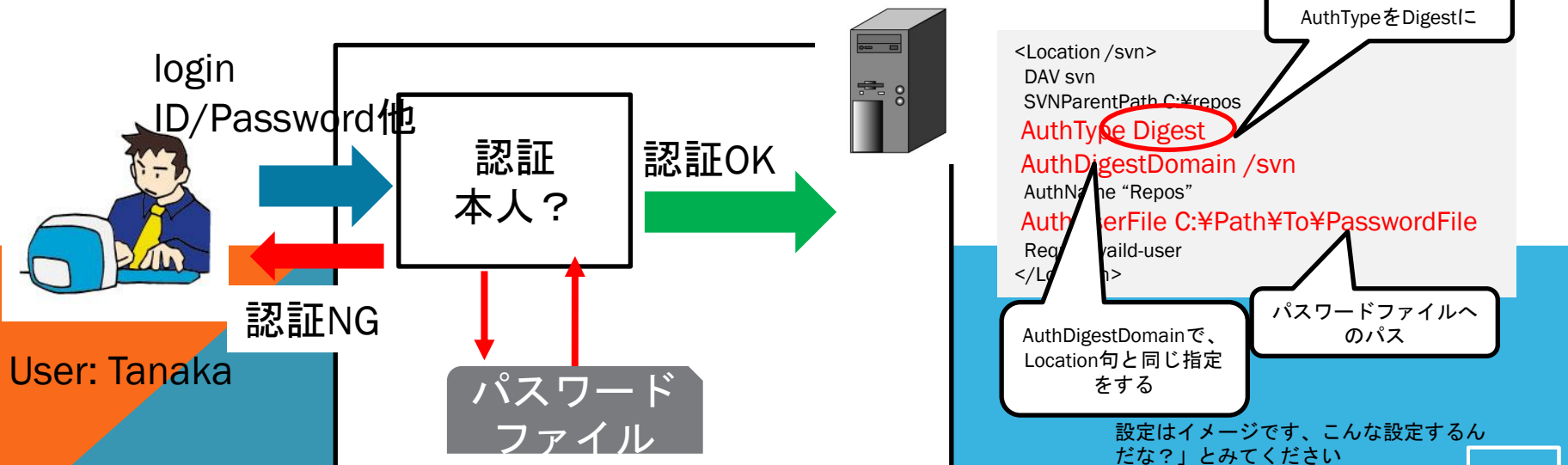


3. Subversionの認証認可

3.1 Subversionの認証

- ダイジェスト認証

Apacheベーシック認証と同様にパスワードファイルを使用する。
Apacheベーシック認証ではパスワードが平文で流れるが、ダイジェスト認証では流れないように工夫されている。



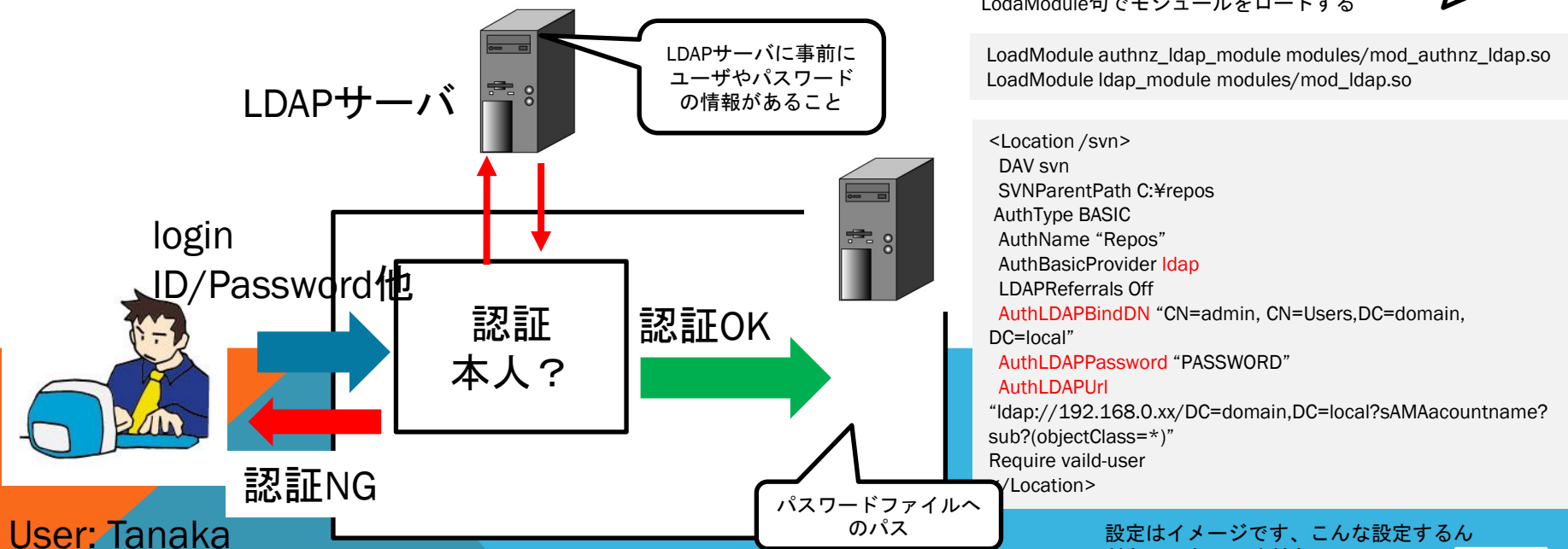
3. Subversionの認証認可

3.1 Subversionの認証

• LDAP認証

ユーザIDとパスワードの照合をLDAPサーバに問い合わせに行く。
LDAPは、認証でよく使われるもので、ディレクトリサーバとも呼ばれる。マイクロソフトのActive Directoryが有名。OSSで主要なものだと、OpenLDAPや389 Directory Server, OpenDJなどがある。

例ではActiveDirectoryにLDAPでつないでいる



3. Subversionの認証認可

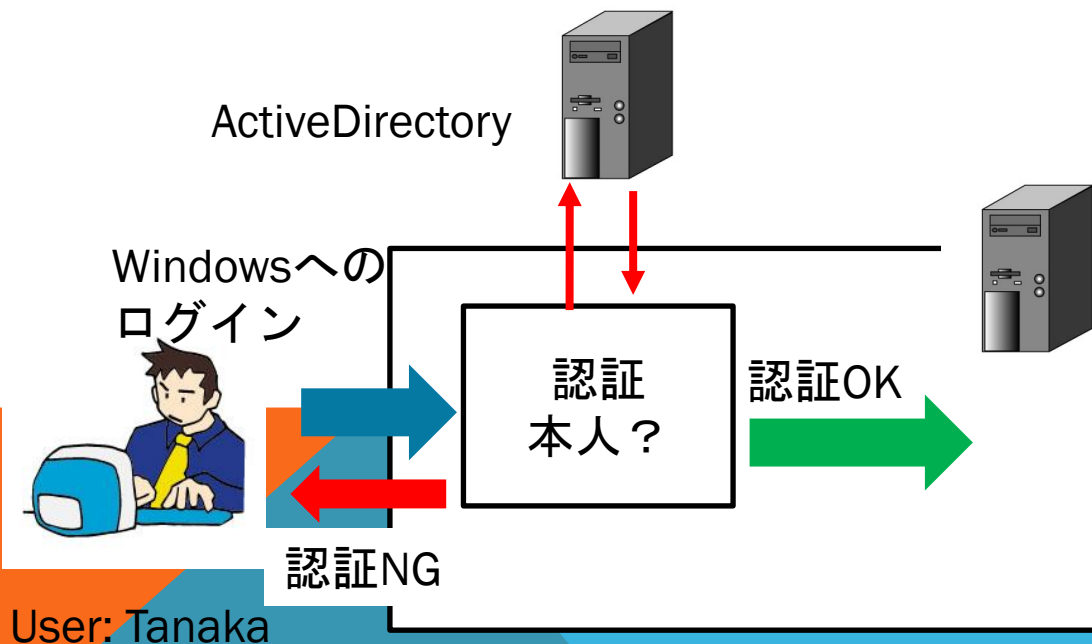
3.1 Subversionの認証

- 統合Windows認証

マイクロソフト社のActive Directoryが提供する「Windowsドメイン」の認証を利用する。Subversion公式ではない専用のモジュールが必要となるが、WindowsのPC端末がWindowsドメイン認証していればそのまま認証される。(windowsのみ)

LodaModule句でモジュールをロードする

```
LoadModule authnz_sspi_module modules/mod_authnz_sspi.so
```



```
<Location /svn>
AuthName "SSPI"
AuthType SSPI
SSPIAuth On
SSPIOfferSSIP On (自動ログイン)
SSPIOfferBasic On (未認証の場合ダイアログ出す)
SSPIBasicPreferred Off (優先設定)
SSPIDomain xxxx.example.com
SSPIOmitDomain On (Case変換)
SSPIPerRequestAuth Off (リクエスト毎の認証)
DAV svn
SVNParentPath C:\repos
SVNListParentPath on
SVNIndexXSLT "svn-web/view/svn.xls"
<RequireAny>
Require all denied
Require valid-user
#Require sspi-group svn_admin_group
<RequireAny>
</Location>
```

SSPIはWindowsで使われるセキュリティ機能

設定はイメージです、こんな設定するんだな？」とみてください

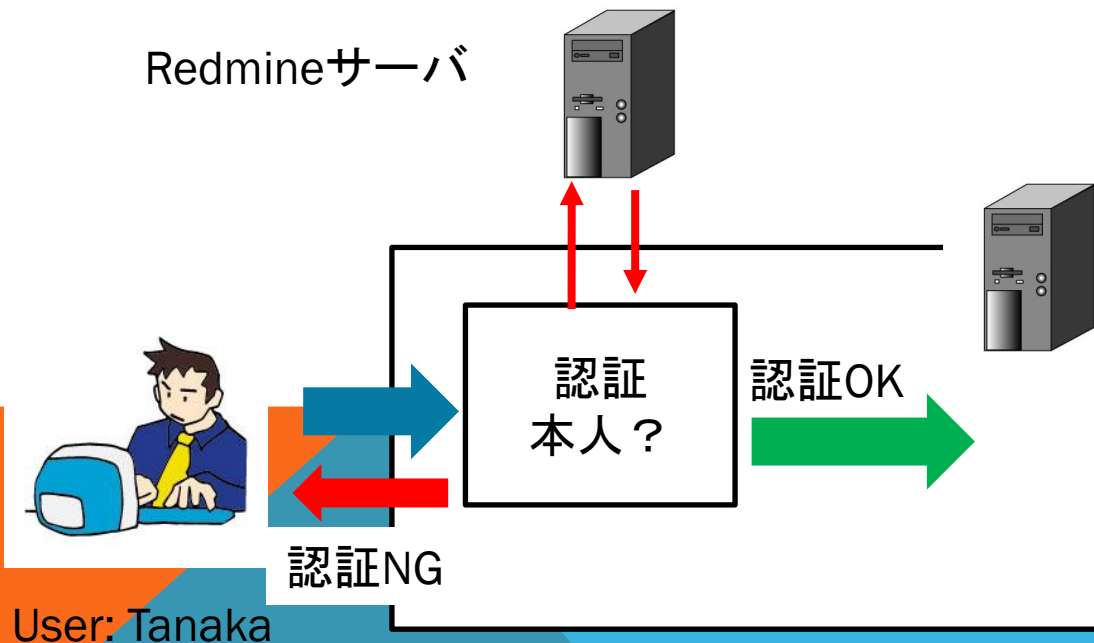
3. Subversionの認証認可

3.1 Subversionの認証

- その他（ちょっと変わったやつ）

この例ではRedmineの認証を使って認証をします。リクエストが来ると、Redmineに問い合わせします。

Redmineのプロジェクト毎の認可（アクセス可否）ができます。



```
PerlLoadModule Apache::Authn::Redmine
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

```
<Location /svn/>
PerlAccessHandler Apache::Authn::Redmine::access_handler
PerlAuthenHandler Apache::Authn::Redmine::authn_handler
DAV svn
SVNParentPath /path/to/repo_parent
SVNListParentPath on
AuthzSVNAccessFile /path/to/svnauthz
AuthType Basic
AuthName Subversion
AuthUserFile /dev/null
RedmineDSN "DBI:Pg:dbname=redmine;host=redmine-pod"
RedmineDbUser "dbuser"
RedmineDbPass "Password"
Require valid-user
</Location>
```

設定はイメージです、こんな設定するんだな？」とみてください

3. Subversionの認証認可

3.2 Subversionの認可

- Subversionは、認証を「リポジトリ毎」でも「リポジトリ群（リポジトリのあるフォルダの上位フォルダ）」でも設定可能。
- なので、リポジトリ毎に認証を設定することで実質的に「リポジトリ毎のアクセス管理」（アクセス可否のみ）は可能
- それ以外の方法として、“Authz”(認可)ファイルを使うことで、「フォルダやファイルレベルのきめ細かい認可設定」が可能になる。

3. Subversionの認証認可

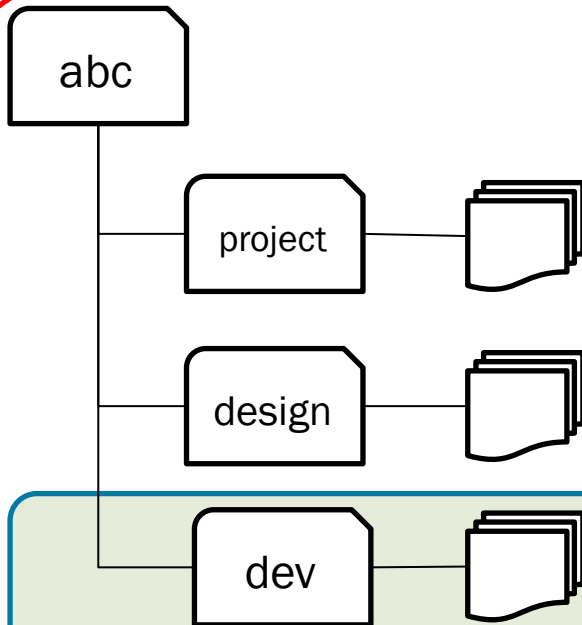
3.2 Subversionの認可

LoadModule authz_svn_module modules/mod_authz_svn.so

念のためモジュールをロード

AuthzSVNAccessFile C:\path\to\sample.authz

Httpd.confやsubversion.confでリポジトリ設定の際、認可ファイルのパスを指定



```
[group]
member = @developers, @testers, @designers
developers = ichiro, jiro
testers = saburo, shiro
designers = goro, rokuro
```

```
[/]
* =
```

```
[abc::/]
@member = r
```

```
[abc::/dev]
@developers = rw
```

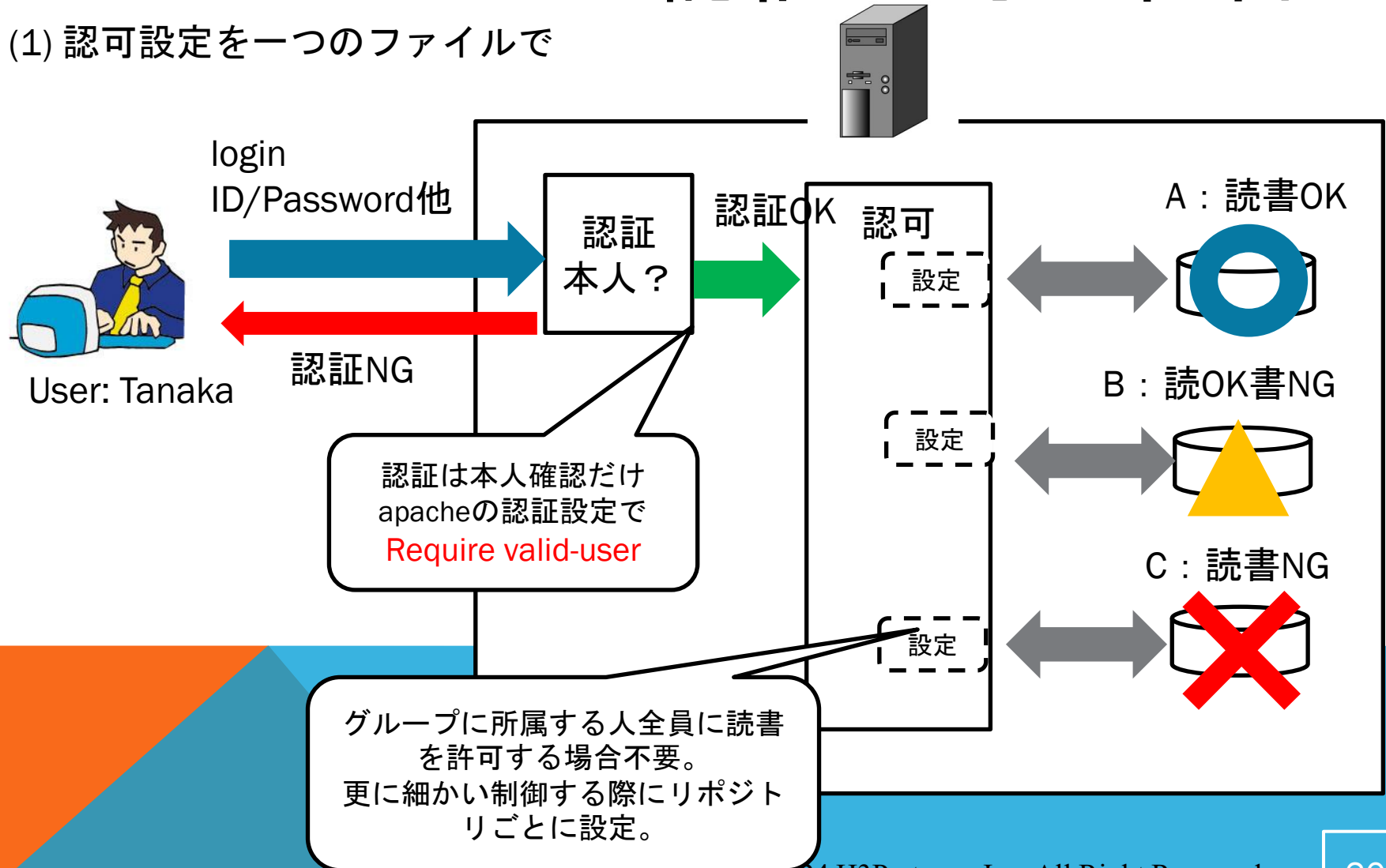
リポジトリabcに対して、プロジェクトメンバー”member”にRead権限

メンバーのうち、開発チーム”developers”には、devフォルダの読み書きの権限を付与

3. Subversionの認証認可

3.3 Subversion認証認可の組合せ

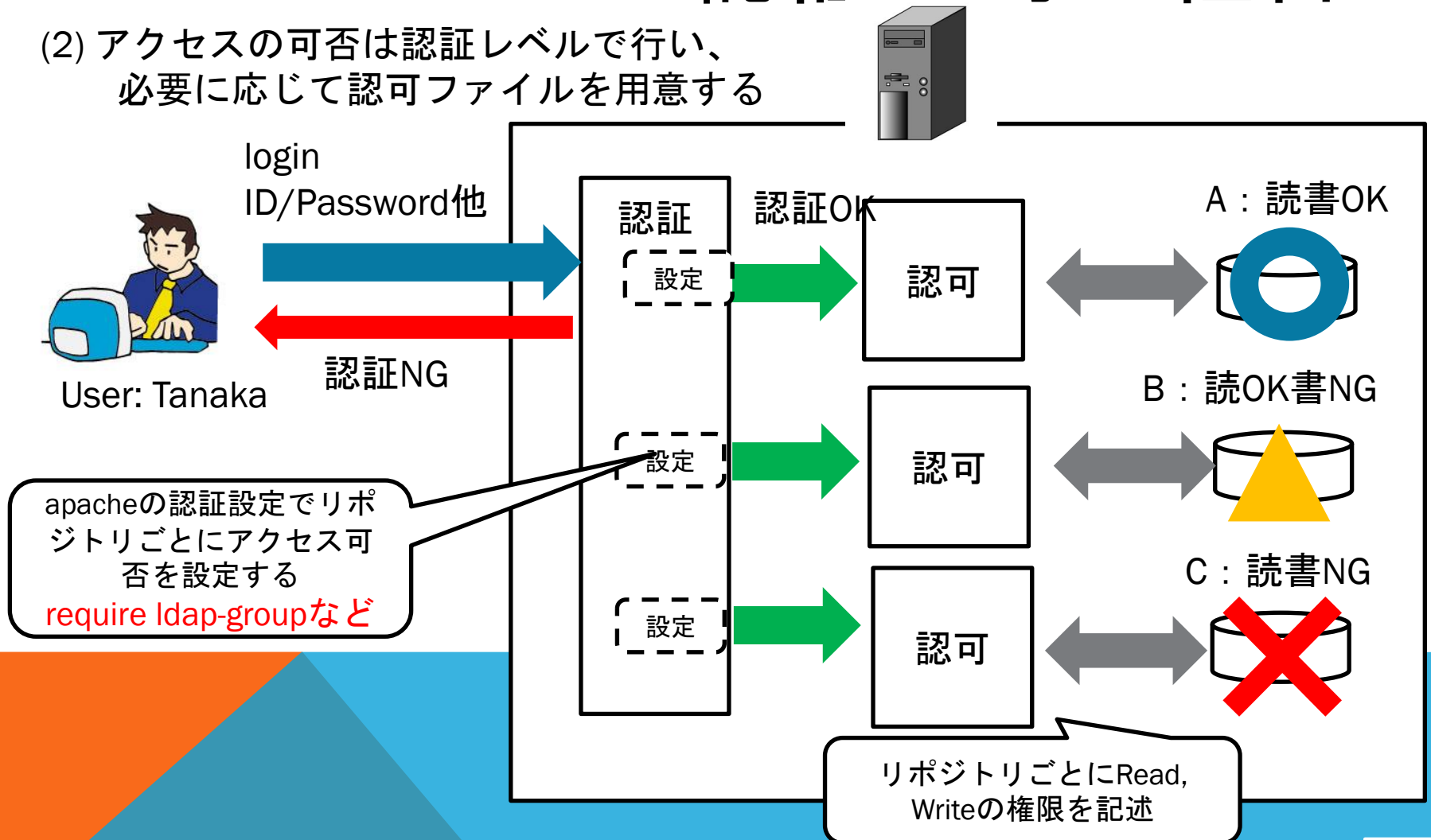
(1) 認可設定を一つのファイルで



3. Subversionの認証認可

3.3 Subversion認証認可の組合せ

(2) アクセスの可否は認証レベルで行い、
必要に応じて認可ファイルを用意する



4. Gitの認証認可

4.1 Gitの認証と認可ssh編

- Gitの接続方法の違いと認証認可機能違い

	アクセスの種類	概要	認証	認可(Read)	認可(writeの)
1	ローカルアクセス	共有フォルダなどにあるGitリポジトリを直接見に行く。	ローカルの認証を信用	OSのアクセス権	デフォルト可(スクリプトで制御)
2	Gitデーモン	gitプロトコル git://hostname/xxxx	認証しない	誰でも参照可	デフォルトでOFF
3	ssh接続	ssh://hostname/xxxx	OSの認証機能	OSユーザのアクセス権	デフォルト可(スクリプトで制御)
4	http接続	https://hostname/xxxx	apacheの認証機能	認証ユーザ	デフォルト可(スクリプトで制御)

4. Gitの認証認可

4.1 Gitの認証と認可ssh編

sshではサーバOSにユーザを作り、ユーザを「リポジトリのオーナーと同じグループ」に所属させる。

ssh://192.168.0.xx:/home/gitrepos/repo01

yusuke



①Git サーバ上に
tanakaユーザを作る

Gitサーバ

tanaka

②tanakaユーザに
公開鍵認証を設定する

sshのパスフレーズは聞かれるが
tortoiseGitだと一度入れると覚えている

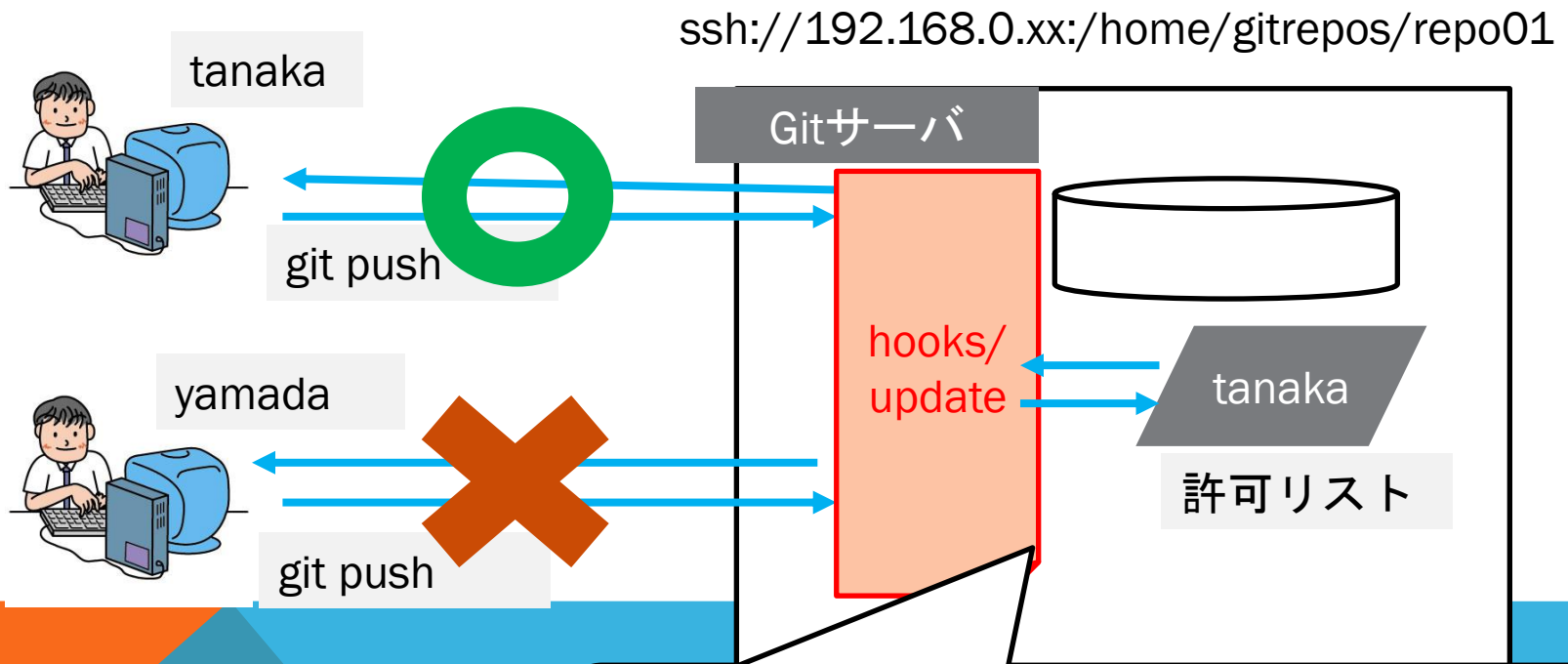
③tanakaユーザを
gitリポジトリの
オーナーと同じグ
ループに入れる

課題：全員書き込みもできてしまう

4. Gitの認証認可

4.1 Gitの認証と認可ssh編

- 書き込みは制限はupdateスクリプトで



ユーザがtanakaだと書き込み許可、でなければ拒否する（スクリプトなので自由に書ける）
制御にOSのユーザを使うかGitユーザ名を使うかは選択可。

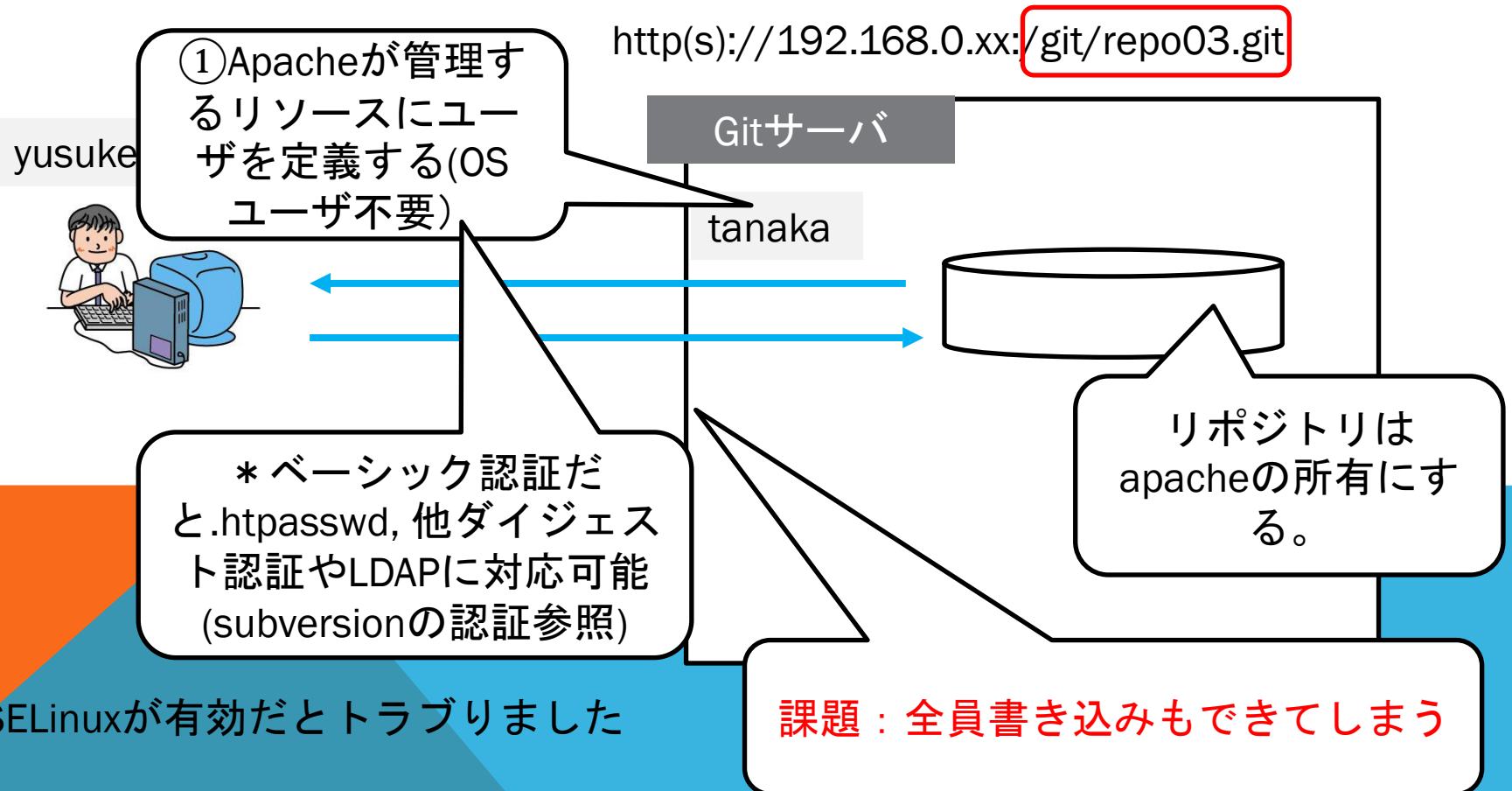
【参考】

<https://maku77.github.io/p/f84ekdg/>

4. Gitの認証認可

4.2 Gitの認証と認可http(s)編

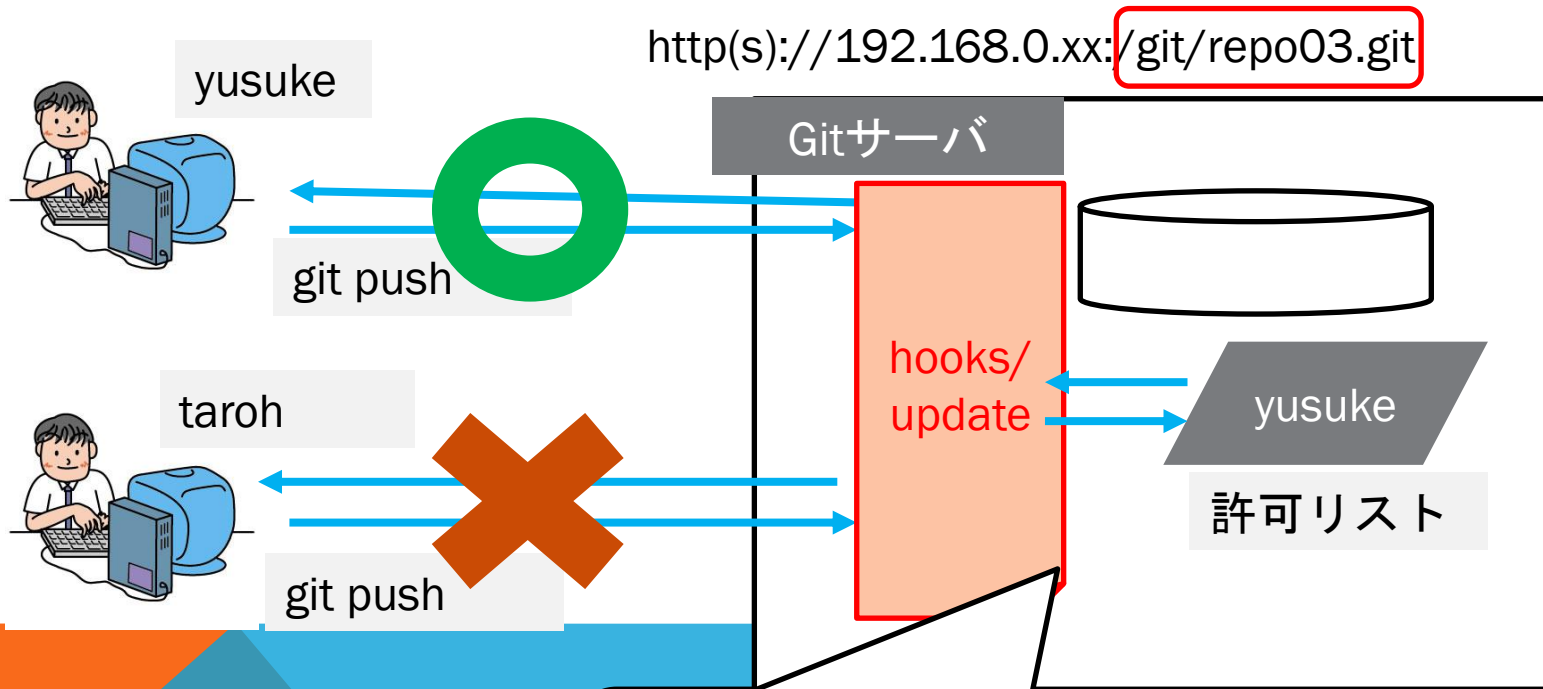
http(s)の場合ApacheからGitを呼び出す



4. Gitの認証認可

4.2 Gitの認証と認可http(s)編

- 書き込みは制限はupdateスクリプトで



【参考】

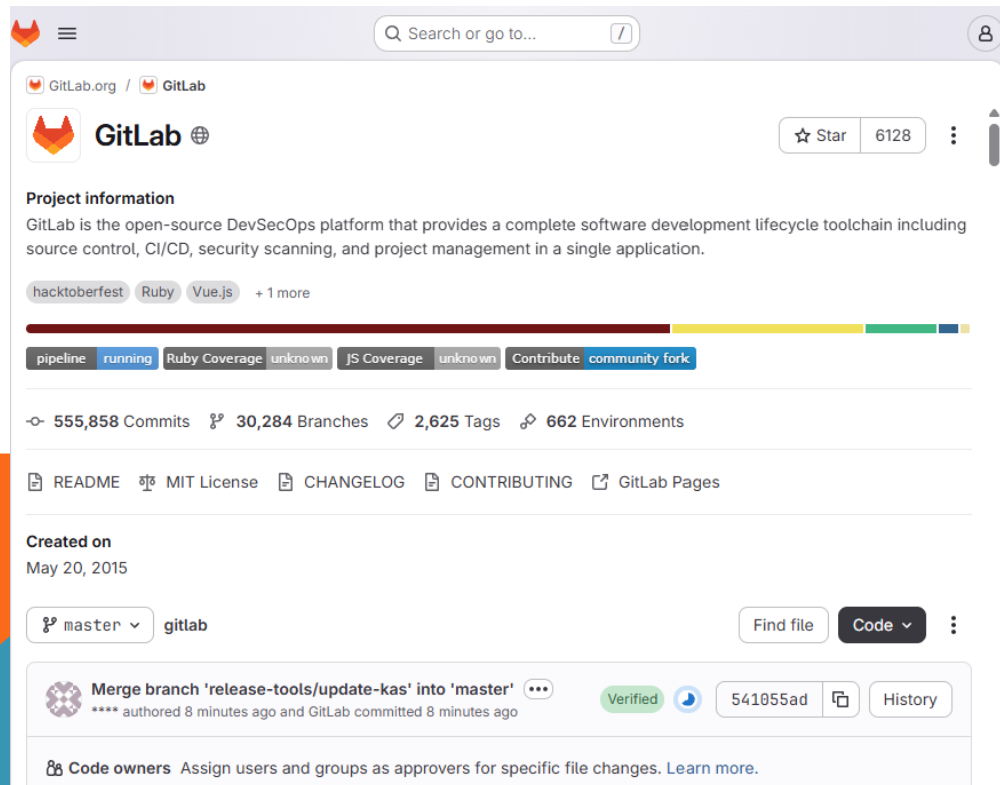
<https://maku77.github.io/p/f84ekdq/>
ユーザ情報の取得方法は変更
\$ git log -1 -format='%an' "\$3"

基本的にはsshと同様hooks/updateを書くが、
OSユーザはapacheになるので使えない。
Gitユーザ名で制御する

4. Gitの認証認可

4.3 Gitその他(GitLab)

- Git公式のマニュアルにも記載されている
プロジェクト、グループなどの概念があって、GUIで操作できるので便利。Git単独で入れるよりも簡単かも・・・。
<https://gitlab.com/gitlab-org/gitlab>



5. Git3.0 (認証認可には関係なさそう)

5.1 Git3.0での変更点

普及しているGitは2.x。長年これできたが、いよいよ3.0が今年の末ごろに出るらしい。ただ、「ユーザ」はほぼ気づかないレベル。

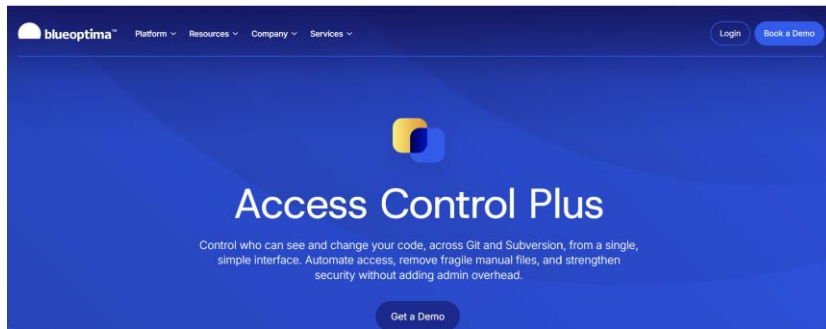
ソース：<https://jsdevspace.substack.com/p/git-30-finally-retires-master-what>

- デフォルトブランチ名がmasterからmainに変更される
- デフォルトのハッシュアルゴリズムがSHA-1からSHA-256に変更される
- object storageが変更（Win/Macユーザに恩恵）
- Rustを正式に取り込むので、ビルドする場合Rustも使用する。

6. BlueOptimaのソリューション

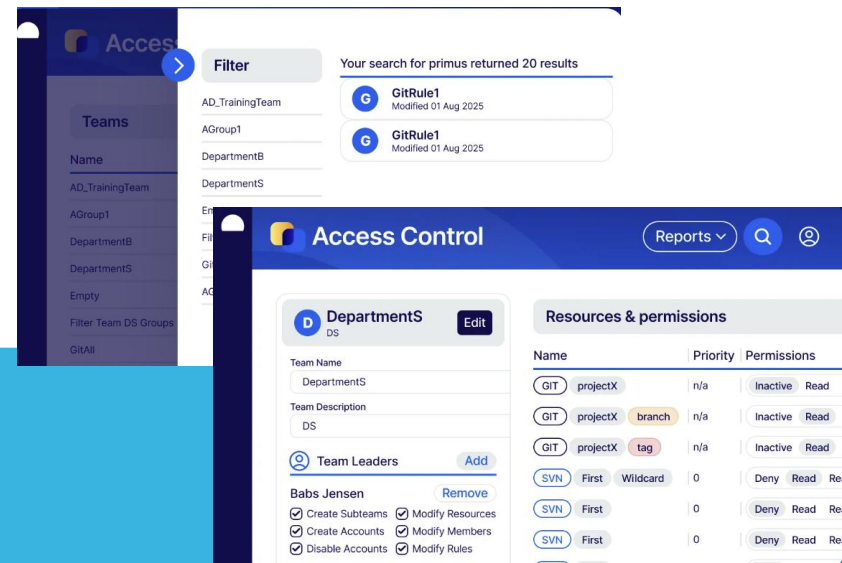
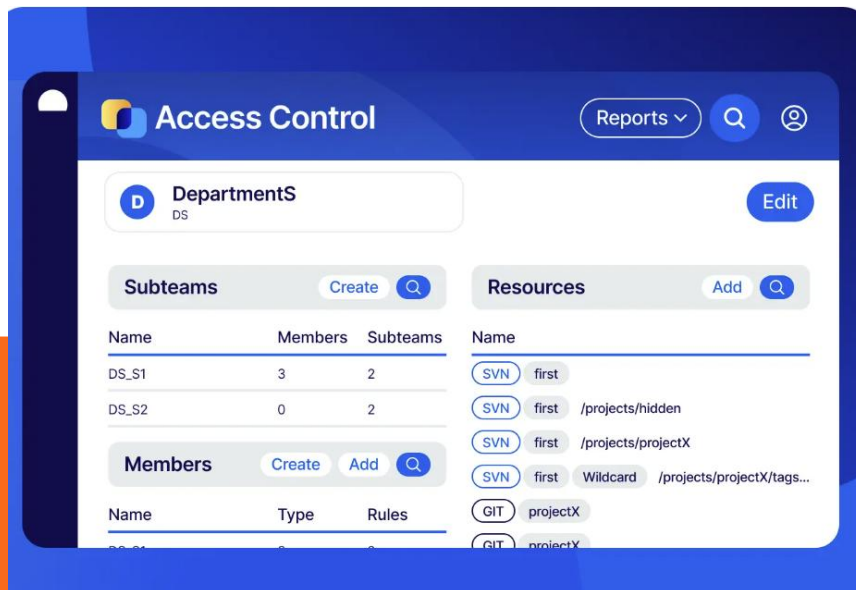
6.1 BlueOptima Access Control Plus

<https://www.blueoptima.com/platform/multisite/access-control-plus>



英国BlueOptima社のツール

複数のGit, Subversionの認証認可をまとめてGUIで管理でき、監査用のログなども出力できる。

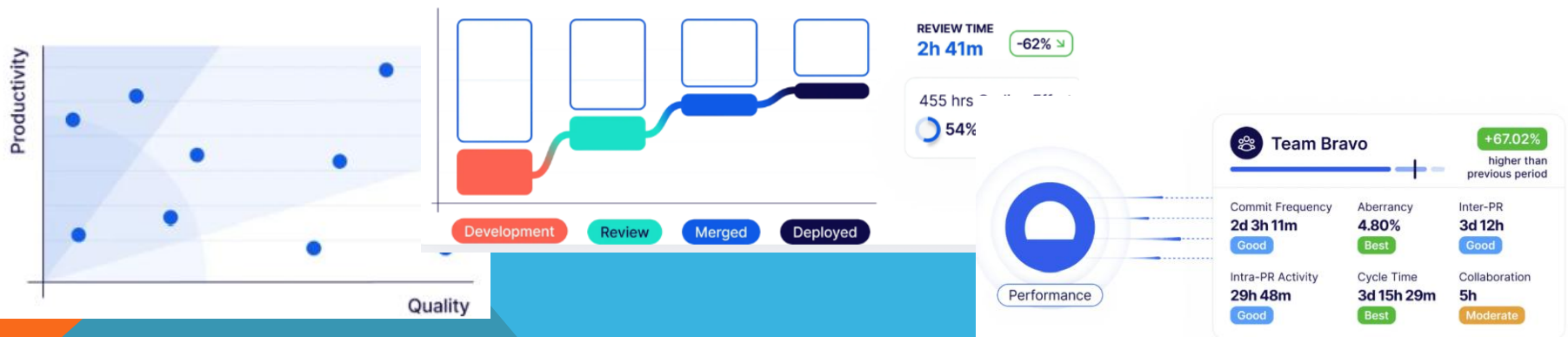
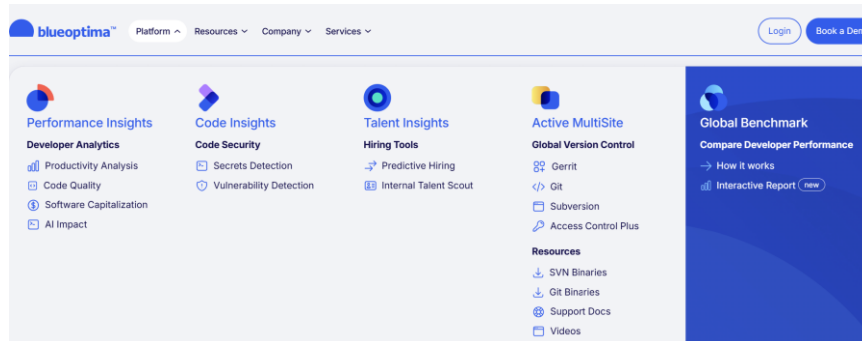


6. BlueOptimaのソリューション

6.2 BlueOptima

英国BlueOptima社は、「コードの生産性」をテーマに開発者や開発チームの生産性を測定・分析するツールを販売している。その際、GitやSubversionなどのリポジトリのコミット状況をユーザ毎、プロジェクト毎、チーム毎などの切り口で分析できる。コードの品質やハードコードされたクレデンシャル、AI作成

の検出ツールなどもあります。



THANK YOU!!!

ありがとうございました