Red Hat
Enterprise Linux

Red Hat Enterprise Linux 8 の Web Console（Cockpit）

森若 和雄
Solution Architect
2019-09

Red Hat

このスライドの位置付けと目的

- 対象
  - GUIでLinuxを管理できると嬉しい人
  - Linuxのデスクトップ環境を持ってないのでXが必要なGUIツールを使うのは難しい人
- 目的
  - RHELに同梱されていて各種サーバ管理に利用できるWeb UIを提供するCockpitを紹介

Red Hat

概要

- Web Console (Cockpit)とは?
- Cockpit Packageによる機能拡張
- Cockpitのセキュリティ

Red Hat

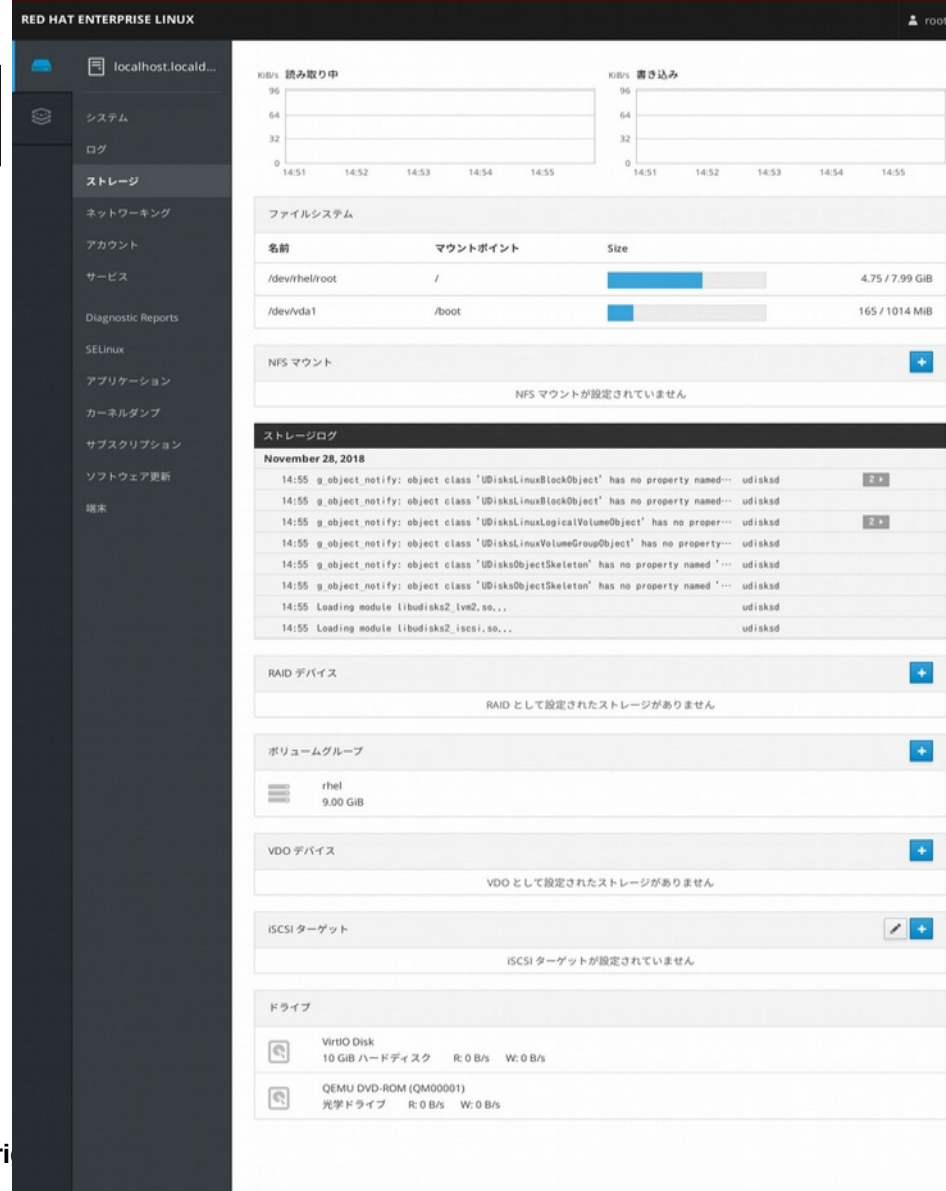
Web Console

COCKPIT

サーバ管理用のWeb UIを提供

- サービス管理
- アカウント管理
- ネットワーク管理
- ファイアウォール管理
- ストレージ管理
- ログ閲覧
- 仮想マシン管理
- 仮想端末　などの管理機能を提供

https://cockpit-project.org/



Cockpitの特長

- 独自のDBや設定ファイルなどをほぼ持たない
  - dbus、ファイルアクセス、コマンド実行などへのアクセスを提供するバックエンドとブラウザからアクセスするjavascript APIを基盤とする
  - アクセス制御はPolicyKitにより実施
- 利用していない時にリソースを事実上消費しない
- プラグイン形式での機能追加が容易
  - ~/.local/ 以下に配置することで特定ユーザのみ利用可能
- Kerberosを利用したSSOに対応

Cockpitのインストール

- パッケージ導入
  - yum install -y cockpit cockpit-dashboard
- 証明書設置
  - cp cockpit.cert /etc/cockpit/ws-certs.d/
- ファイアウォール設定
  - firewall-cmd --add-port=9090/tcp
  - firewall-cmd --permanent --add-port=9090/tcp
- 起動設定
  - systemctl enable cockpit.socket
  - systemctl start cockpit.socket
- ブラウザで接続
  - firefox https://rhel8.corp.example.com:9090/

TLS用にCA局発行の証明書を配置します。なければ自己署名証明書が自動生成されます。

cockpitそのものの設定をしなくても動作します

Red Hat

リモート管理

- cockpitはブラウザが直接接続できないシステムも管理できる (cockpit-dashboardパッケージが必要)
- cockpit同士でsshによる通信をおこなう
  - 踏み台サーバからsshによる接続ができればリモートのcockpitと直接通信できなくても管理できる
  - cockpitへログインしたユーザの権限でssh接続

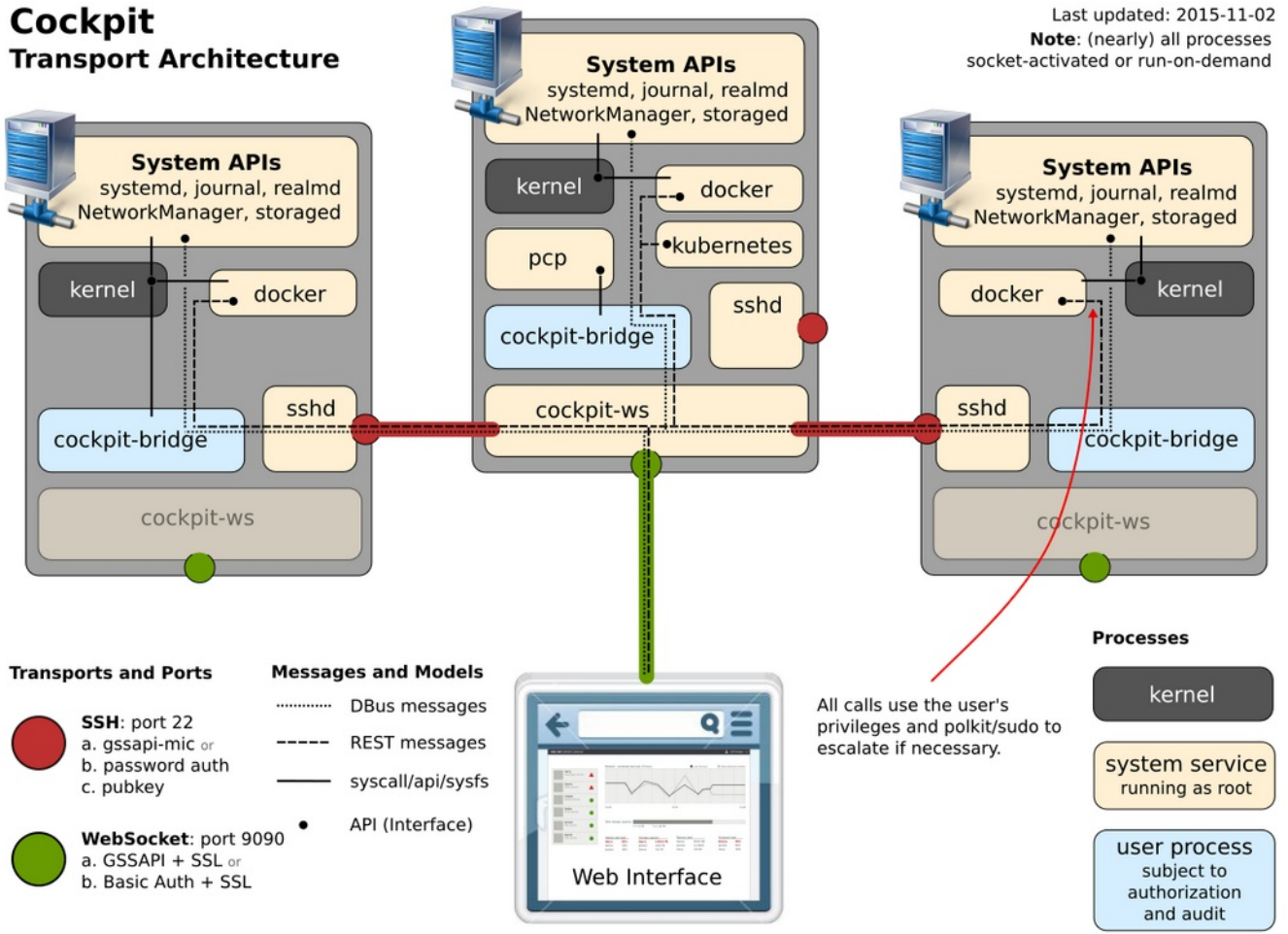
Cockpit Packageによる機能拡張

Red Hat

Cockpitの基本動作

https://raw.githubusercontent.com/cockpit-project/cockpit/master/doc/cockpit-transport.png

- cockpit-wsが待受け
- ブラウザでログイン
- cockpit-bridgeがユーザ権限で起動
- cockpit-bridgeがdbus, pcp, プロセス実行などを行う
- リモートへはsshで接続



Cockpitを拡張するには?

- プラグイン(Cockpit Package)による拡張に対応
- dbusでのリクエストやコマンド実行、ファイル読み出し等を行うjavascript用APIを提供

コマンド実行

```
cockpit.spawn(["ping", "8.8.8.8"])
```

ホスト名取得

```
proxy = cockpit.dbus("org.freedesktop.hostname1").proxy()
```

ファイル内容取得

```
file = cockpit.file("/etc/motd", {})
```


Red Hat

Cockpit Packageとは?

以下のようなファイル群を配置することで**Cockpit**内で利用する画面を作成する。

/usr/share/cockpit/

hoge/	パッケージ用ディレクトリ
manifest.json	メニュー内の場所、 ラベルや関連ファイルなどを指定
hoge.html	表示する**html**
hoge.js	操作を実装する**javascript**

Cockpit Packageの配置場所

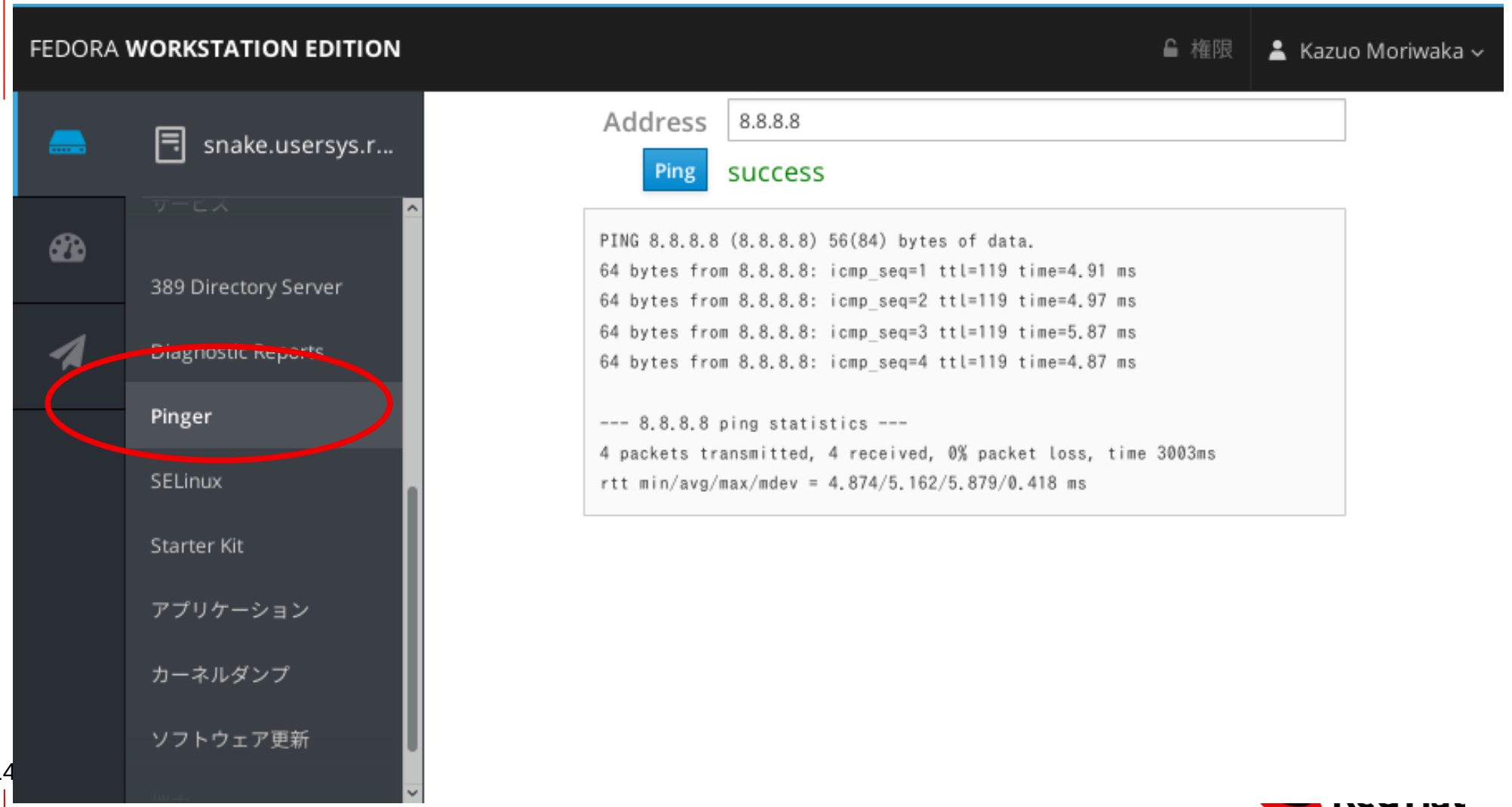
- ファイルを配置する場所が3箇所あり、オーバーライドできる
  1. /usr/share/cockpit/* (rpm用、システム全体用)
  2. /usr/local/share/cockpit/* (システム全体用)
  3. ~/.local/share/cockpit/* (特定ユーザ用)
- cockpit-bridge --packages コマンドで(実行したユーザにとっての) Cockpit Package一覧を表示
- manifest.jsonで同じ名前を宣言した場合、manifest.json内のpriorityの値が大きい方が優先される

Red Hat

個人用にCockpit Packageを作ってみる

- インストール先ディレクトリ作成
  - mkdir -p ~/.local/share/cockpit/
- cockpitのソースコードを取得
  - git clone https://github.com/cockpit-project/cockpit.git
- シンボリックリンク作成
  - cd cockpit/examples/
  - ln -s ${PWD}/pinger ~/.local/share/cockpit/
- ブラウザでcockpitへ接続(または再ログイン)
  - メニュー上の“Pinger”メニューでpingコマンド実行

Red Hat



Cockpit Packageの 参考文献

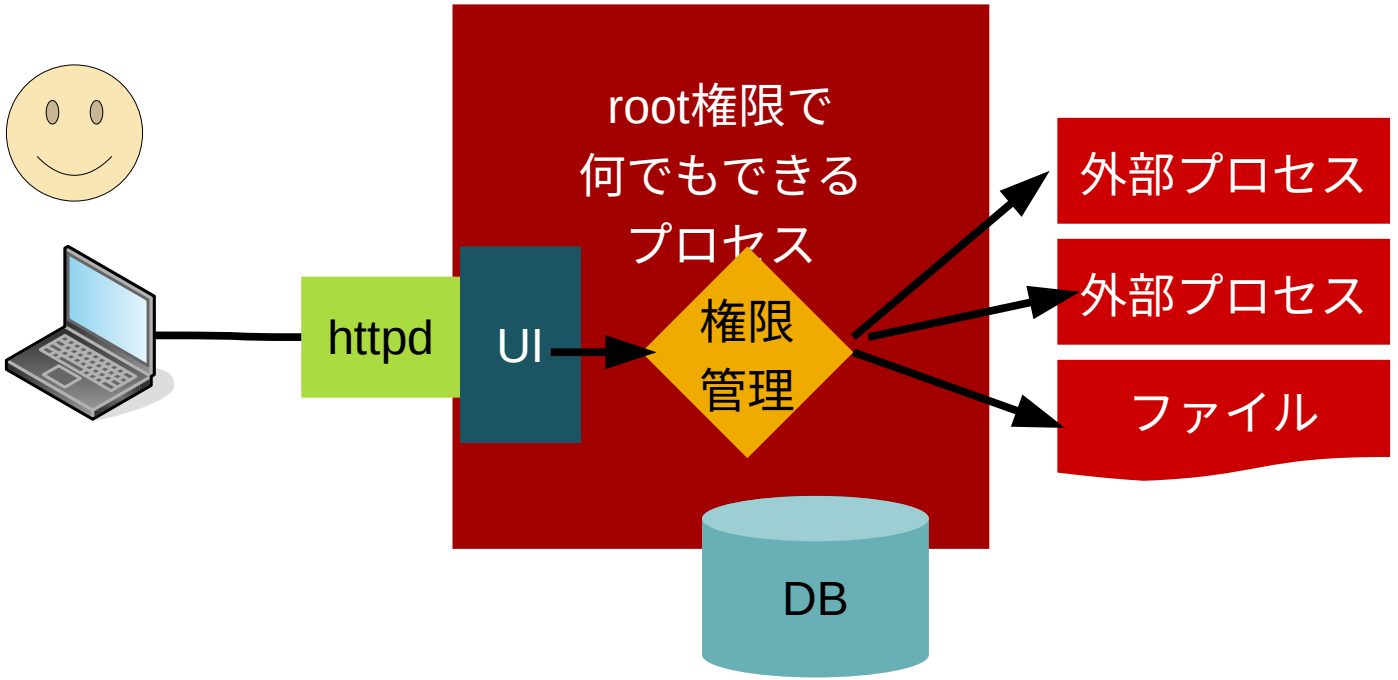
- cockpit projectのTutorialカテゴリ
  - https://cockpit-project.org/blog/category/tutorial.html
- cockpitプロジェクトのソースコード
  - https://github.com/cockpit-project/cockpit/
- 本格的なpluginのひな型 “Starter Kit”
  - rpmパッケージ化、テストケース実行, 一般公開に対応
  - https://github.com/cockpit-project/starter-kit
- Cockpitドキュメント内 “Developer Guide”
  - https://cockpit-project.org/guide/latest/development.html

Red Hat

Cockpitのセキュリティ

Red Hat

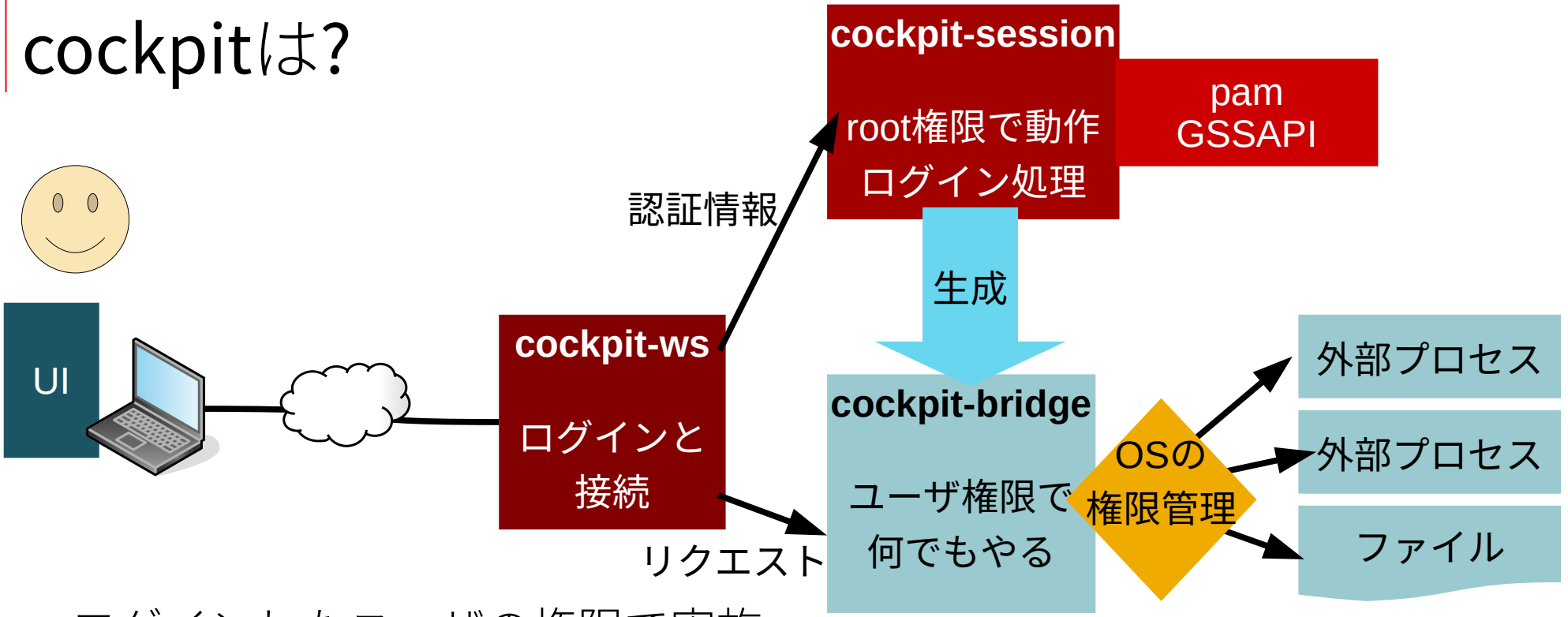
よくある(?)「Webコンソール」のイメージ(偏見)



- 外部コマンド実行やファイル変更をroot権限で実施
- 「root権限で何でもできるプロセス」
  - 複雑になり、バグが深刻な問題につながる
- 独自の権限管理設定やDBが必要

Red Hat

cockpitは?

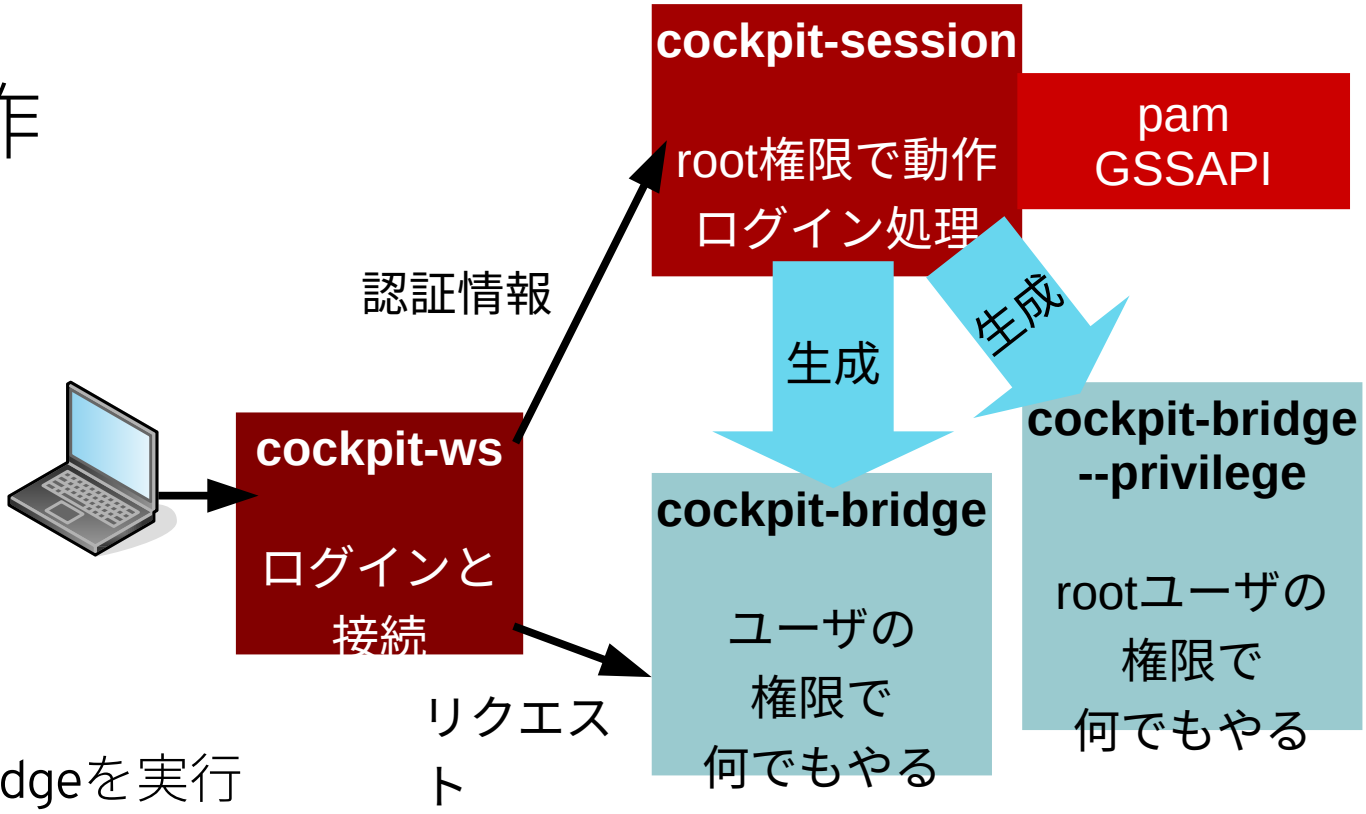


- ログインしたユーザの権限で実施
- root権限利用はセッション開始時のみ
- OSの権限管理をそのまま使う

Red Hat

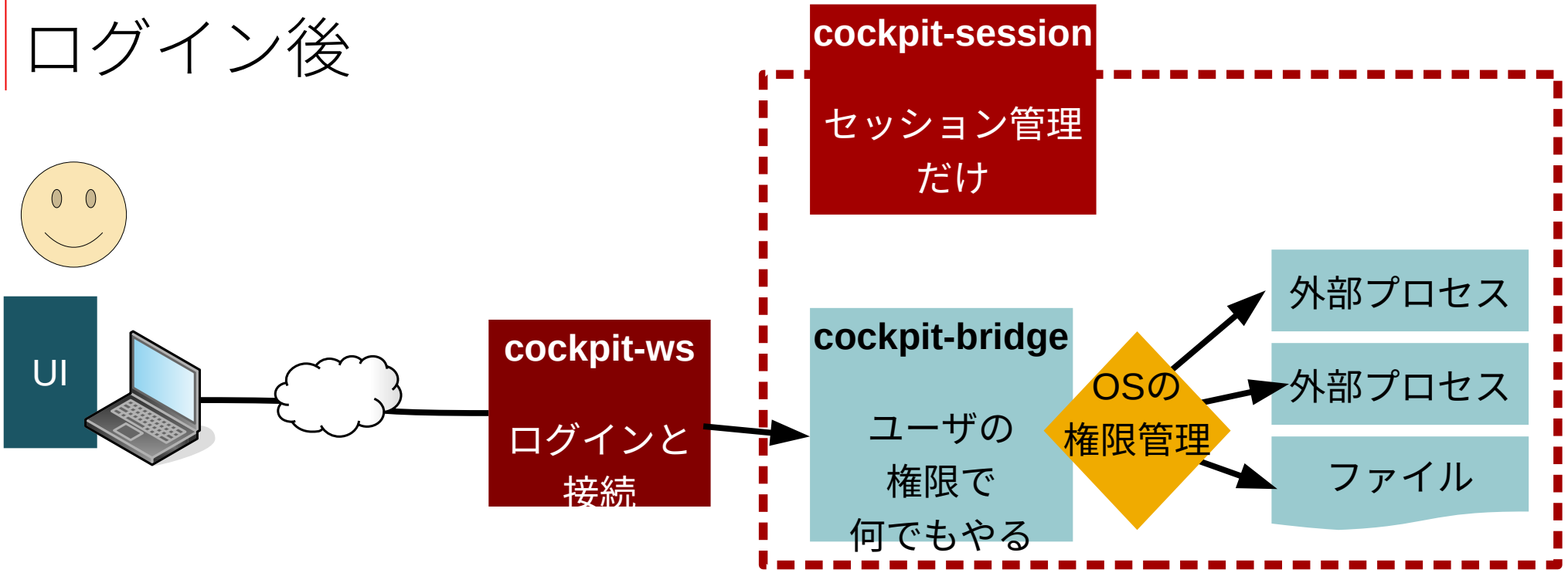
ログイン時の動作

1. cockpit-wsがログイン画面を表示
2. 認証情報を送付
   - ID+Password
   - GSSAPI
3. cockpit-sessionを実行
4. PAMで認証
5. 認証ユーザでcockpit-bridgeを実行
6. 以下チェックのある場合pkexecでroot権限のcockpit-bridgeを起動



デフォルトではwheelグループのユーザのみ成功

ログイン後



- ブラウザ上のjavascriptとcockpit-bridge(およびcockpit-bridge --privilege)が通信
- cockpit-bridge本体と、packageと呼ばれるプラグインがdbus, REST API, プロセス実行など各種の操作を実施。通常のユーザ権限管理をそのまま利用。

Cockpitに関連したセキュリティ上の注意点

- SELinuxのポリシーも準備されているのでできれば使う
  - enforcingで使うことでcockpit-wsやcockpit-sessionに未知の脆弱性があった場合にも任意プロセス実行などを予防する
- 自己署名証明書の利用はお勧めしません
  - ブラウザとcockpit-ws間の通信路が信頼できてはじめて安全な仕組み
  - 後述のIdentity Managementとの統合で証明書の自動発行が可能
- 特権取得にPolicyKitを利用している
  - systemdでも利用するがサーバ管理時に気にしないケースが多いので注意
  - 一部操作を制限したい場合は /etc/polkit-1/rules.d/*.rules でポリシー定義をおこなう。ポリシー制限の粒度はサービスによる。

Identity Managementとの統合

- RHEL同梱のRed Hat Identity Managementと統合されています
- IdMが管理するドメインにサーバが参加していれば以下が可能
  - Web Consoleのリモート管理で利用するssh接続をシングルサインオン(SSO)で処理
  - Web Consoleで利用するサーバ証明書を自動発行し自己署名証明書の利用なし

Red Hat

まとめ

- RHELの管理を簡単にするWeb Consoleは、簡単に利用できるだけでなくセキュリティにも配慮して開発されています
  - ログイン時に最小の特権だけを利用
  - 特権を利用する範囲についてSELinuxで制限が可能
  - 実績があるPolicyKitなど枯れた技術を活用
- Identity Managementと統合されています

Red Hat

Cockpitのセキュリティ 参考文献

- RHEL8ドキュメント「Web コンソールを使用したシステムの管理」
  - https://bit.ly/rhel8-webconsole-ja2
- Cockpitプロジェクトblog「Is Cockpit Secure?」
  - https://cockpit-project.org/blog/is-cockpit-secure.html
- Cockpitプロジェクトドキュメント「Single Sign On」
  - https://cockpit-project.org/guide/latest/sso.html

Red Hat

Thank You

Red Hat