UiPath 製品のバージョンアップガイド - アクティビティ編

2020年3月
UiPath 株式会社

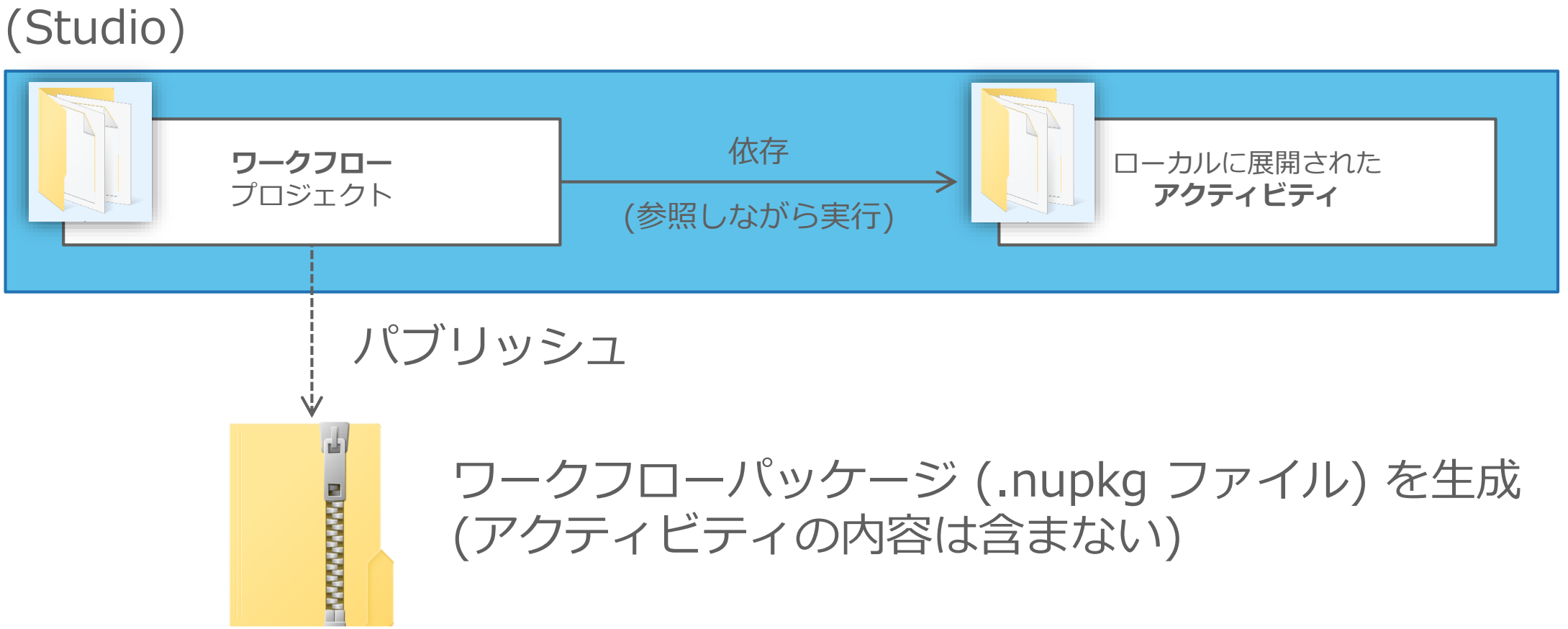
目次

- パッケージファイルの展開と実行
  - 前提となる製品動作を理解する
- Dependency per Project
  - アクティビティバージョンに起因する問題とその解決を理解する
- Core アクティビティのデカップリング
  - Core アクティビティの製品本体からの分離を理解する
- アクティビティの管理とフィード設計指針
  - 上記理解の基、アクティビティパッケージの管理とフィード設計を理解する

パッケージファイルの展開と実行

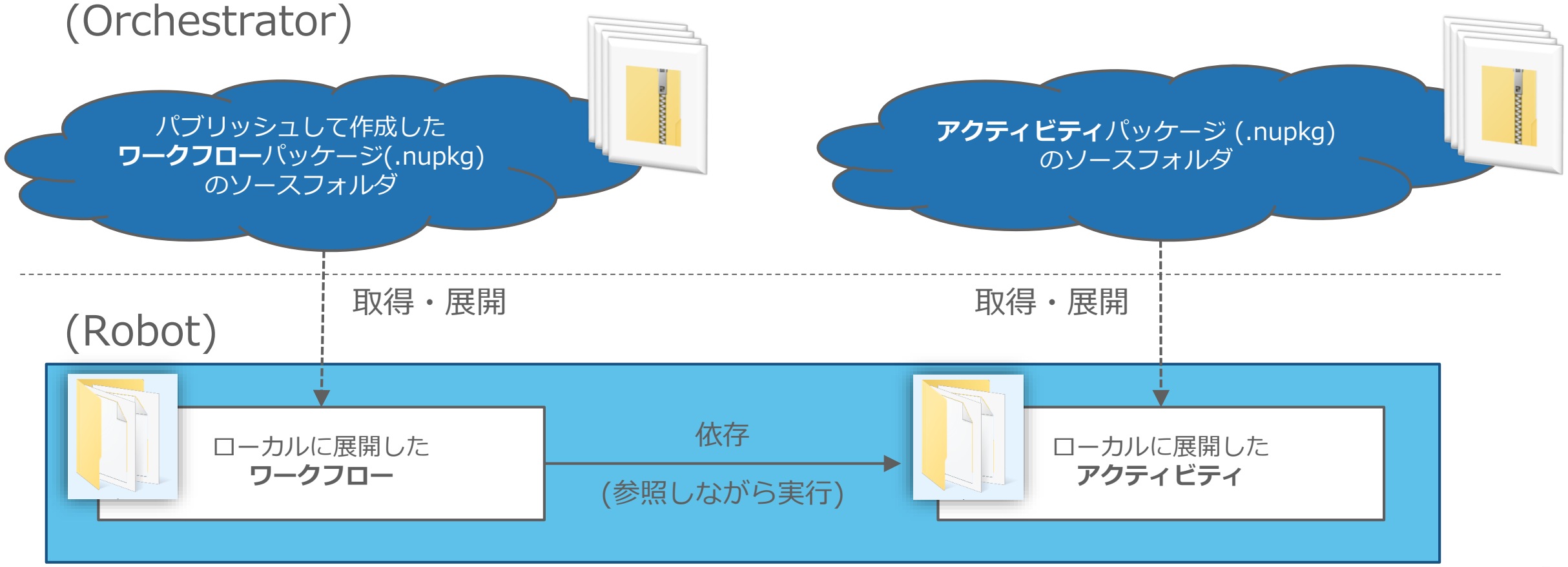
Studio でワークフローパッケージを作成

ワークフローをパブリッシュする事で、 ワークフローパッケージ (.nupkg ファイル) が生成されます。この .nupkg ファイルには、アクティビティは含まれません。



Robot でワークフローパッケージを実行

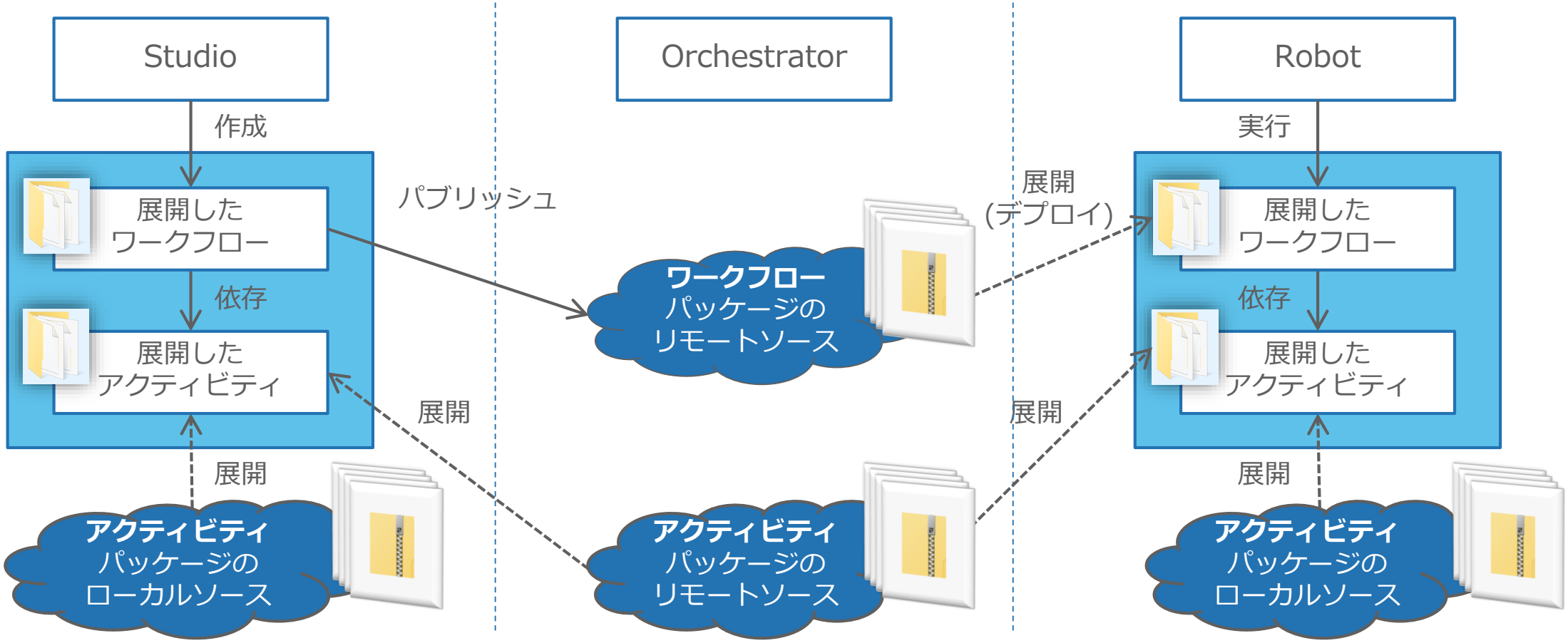
Robot でワークフローを実行するには、作成した際に使用したバージョンのアクティビティが、Robot 端末に展開されている必要があります。



パッケージの展開と実行

Orchestrator にパブリッシュした際の関係図です。

パッケージの解決の関係図

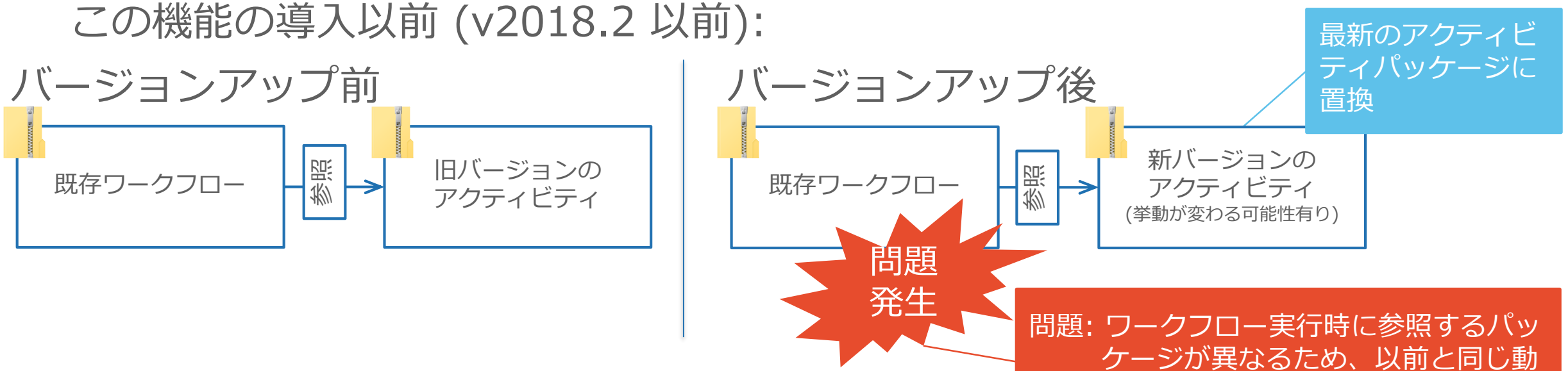


Dependency per Project

Dependency per Project について (1/2)

v2018.2以前の製品バージョンアップ前後で、ワークフロー実行時に参照するアクティビティパッケージが、バージョン差異に起因した挙動の差異により問題が生じることがありました。

この機能の導入以前 (v2018.2 以前):

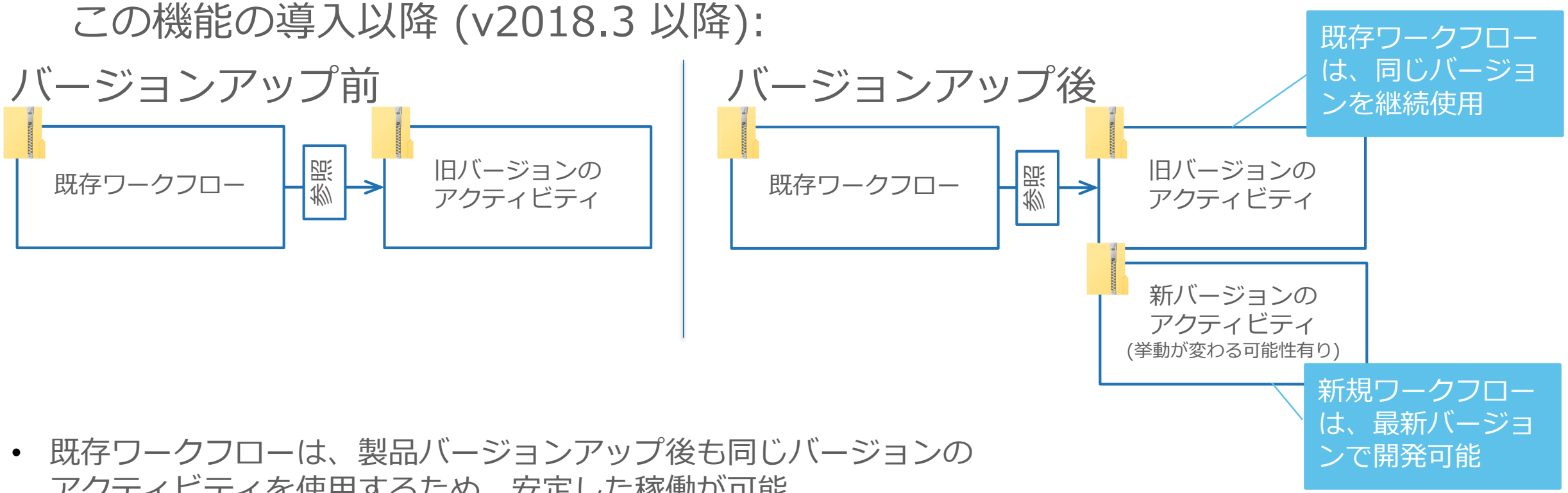


- この問題が発生する可能性があるため、弊社製品のバージョンアップには、特に丁寧な計画・テストが必要
- 製品本体のみバージョンアップし、アクティビティはバージョンアップしないことでこの問題を回避する選択肢もあるが、その場合には新規 WF 開発時にも最新バージョンのアクティビティは使用できない

Dependency per Project について (2/2)

この問題は、v2018.3で導入された新機能により解決します。この機能は日本のお客様からのフィードバックにより実現しました。

この機能の導入以降 (v2018.3 以降):



- 既存ワークフローは、製品バージョンアップ後も同じバージョンのアクティビティを使用するため、安定した稼働が可能
- 一方で、新規ワークフロー開発では新バージョンのアクティビティを利用可能

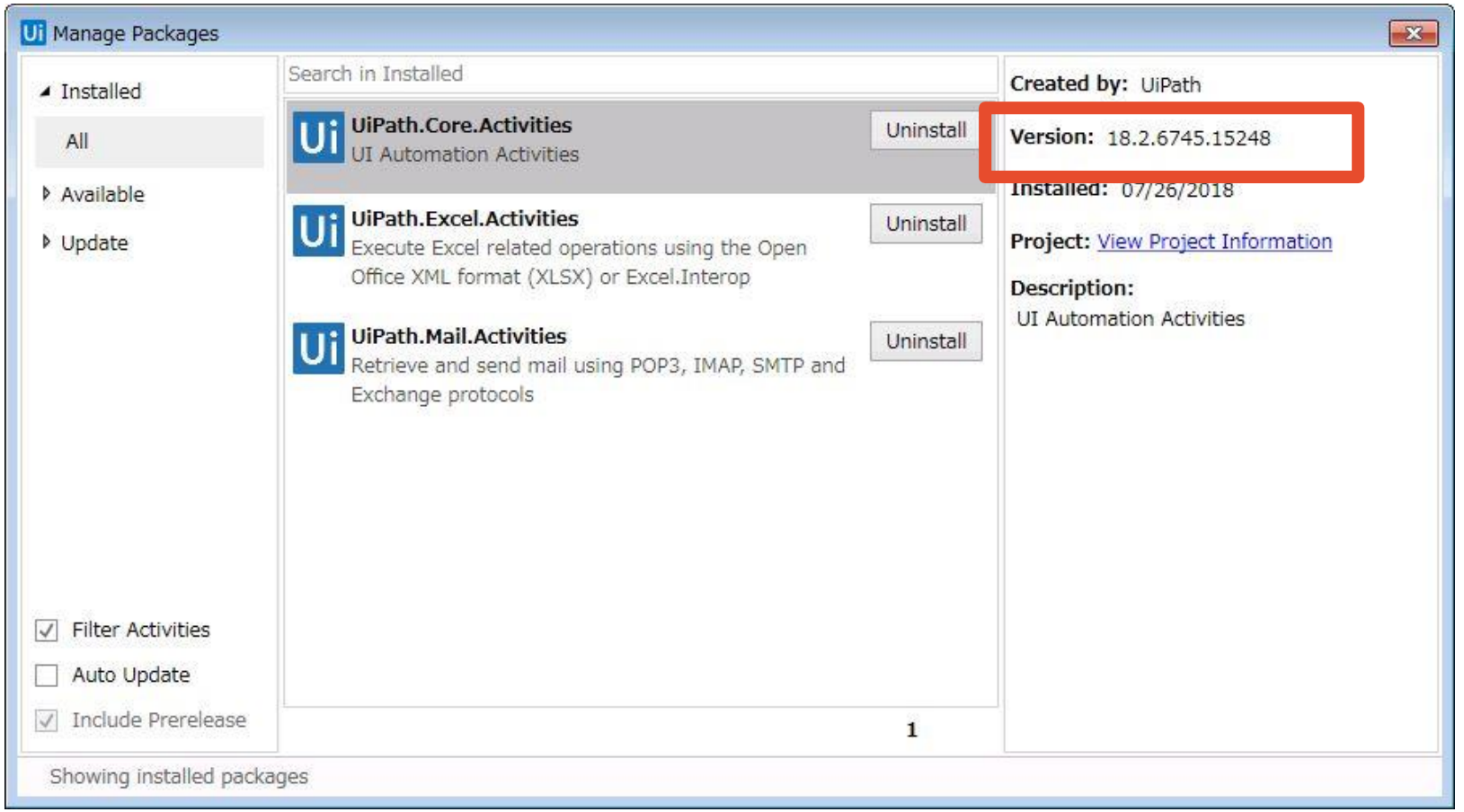
アクティビティパッケージのバージョン解決ルール

v2018.2 以前には、Applicable Version ルールしかありませんでした。
これは後方互換性の維持に問題があるため、v2018.3 で廃止されました。

#	ランタイムルール	製品バージョン	動作	特徴
1	Applicable Version (同じか、より新しいバージョン)	v2018.2 以前	ワークフローで指定したバージョンと同じか、より新しいバージョンであれば、任意のバージョンのアクティビティを呼び出して動作	UiPath 製品バージョンアップ後に、既存のワークフローが壊れるリスクありv2018.3以降では廃止
2	Strict (厳密に同じバージョン)	v2018.3 以降	ワークフローで指定したバージョンのアクティビティパッケージのみを使って動作	UiPath 製品バージョンアップ後にも、既存のワークフローが安定して動作できる安全なルール
3	Lowest Applicable Version (LAV:適用可能な中で最も低いバージョン)	v2018.3 以降	#1 と同じだが、可能な限り低いバージョンのアクティビティを探して動作	後方互換性維持のため、18.2 以前の Studio でパブリッシュしたワークフローを18.3以降のRobotで実行するときに適用されるルール

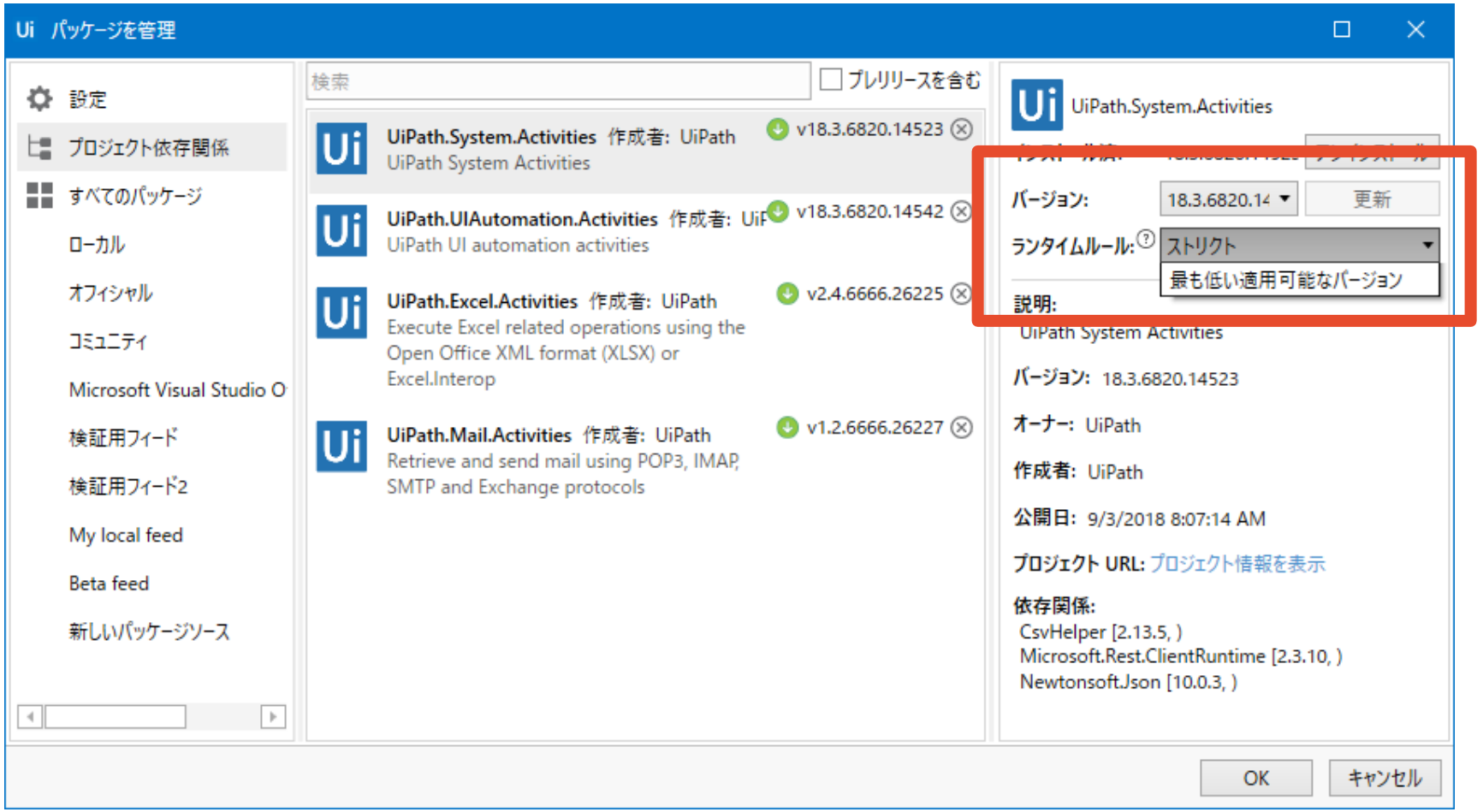
v2018.2 以前でのアクティビティバージョンの指定

アクティビティパッケージは、Studio にインストールされます。この環境で作成したワークフローは、ここで指定されたバージョンのパッケージに対して Applicable Version で動作します。



v2018.3 以降でのアクティビティバージョンの指定

アクティビティは、ワークフロープロジェクトにインストールされます。
パッケージごとに、どのランタイムルールを適用するかを設定できます。



WFを実行時に適用されるランタイムルールまとめ

既存のワークフローを v2018.3 以降で実行する際には、既定で LAV ルールが適用されます。

WF をパブリッシュした Studio のバージョン ＼ WF を実行する Robot のバージョン	**v2018.2 以前**	**v2018.3 以降**
18.2 以前	Applicable Version (equals or higher)	Lowest Applicable Version (LAV)
18.3 以降	本組み合わせは、サポートされません	Strict

- v2018.2 以前の Studio で作成した WF プロジェクトを Studio v2018.3 以降の Studio で開くと、プロジェクトは自動でマイグレーションされ、Strict ルールが適用される
- v2018.3 以降の Studio で作成した WF プロジェクトは、既定で Strict が適用されるが、この設定を手動で LAV への変更も可能 (既定の Strict のままで運用されることを推奨)

Core アクティビティのデカップリングについて

Core アクティビティのデカップリングについて (1/2)

v2018.2 以前では、Core アクティビティは製品本体と密に結合(カップリング)しているため、Core アクティビティについては、Studio/Robot に同梱されたバージョンのものしか利用できません。

Studio/Robot 製品本体

結合

Core
アクティビティパッケージ

一方で、Core 以外のアクティビティについては、任意のバージョンを Studio にインストールして使用可能

Excel
アクティビティパッケージ

Mail
アクティビティパッケージ

Core アクティビティのデカップリングについて (2/2)

v2018.3 以降は、Core アクティビティは System と UIAutomation に分割され、製品本体と分離(デカップリング)されました。そのため、プロジェクト毎に任意のバージョンのパッケージを指定可能になりました。

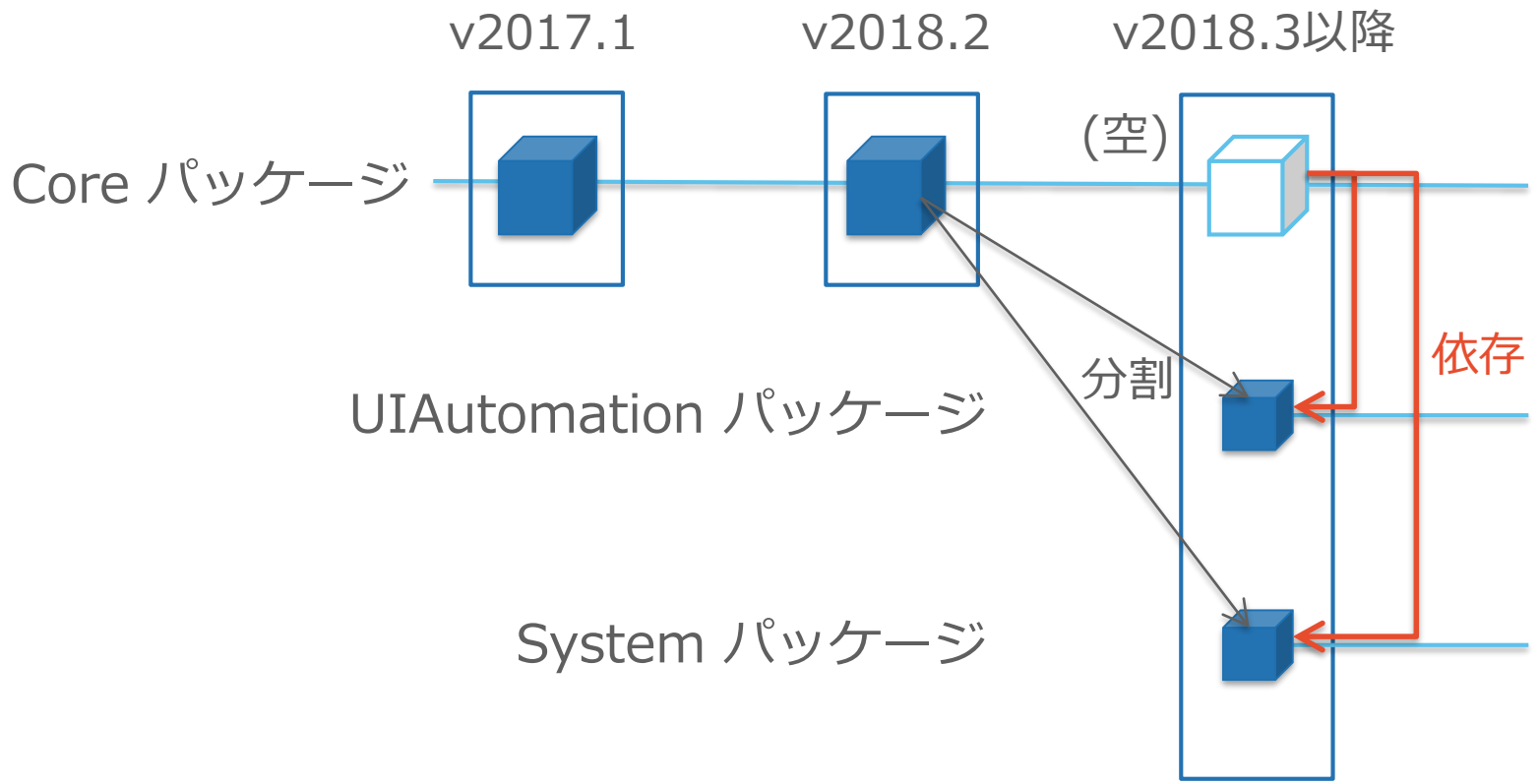
Studio/Robot 製品本体

任意のアクティビティの任意のバージョンを
プロジェクトにインストールして使用可能

- System
  アクティビティパッケージ
- UIAutomation
  アクティビティパッケージ
- Excel
  アクティビティパッケージ
- ...

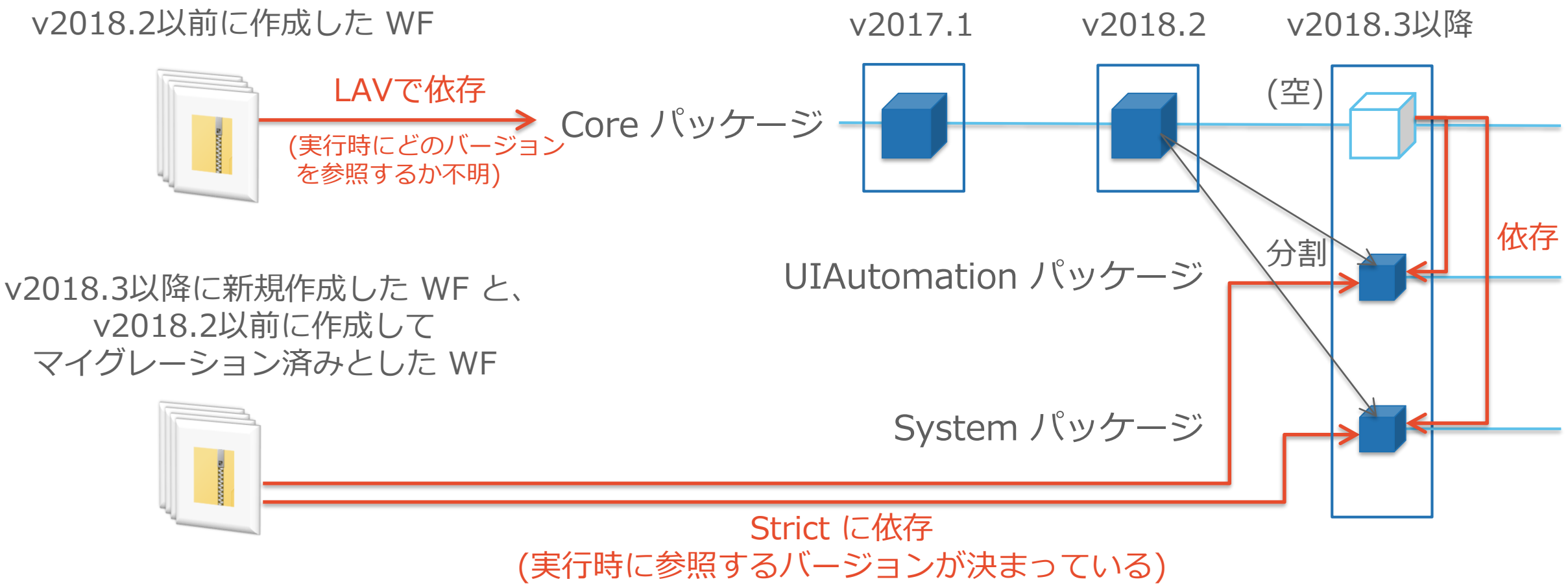
Core アクティビティのバージョン履歴

Core アクティビティは、v2018.3 で System と UIAutomation に分割されましたが、既存WFが参照できるよう空の Core ファイルが残されています。この空のファイルは、System と UIAutomation を参照します。



Core アクティビティのバージョン履歴

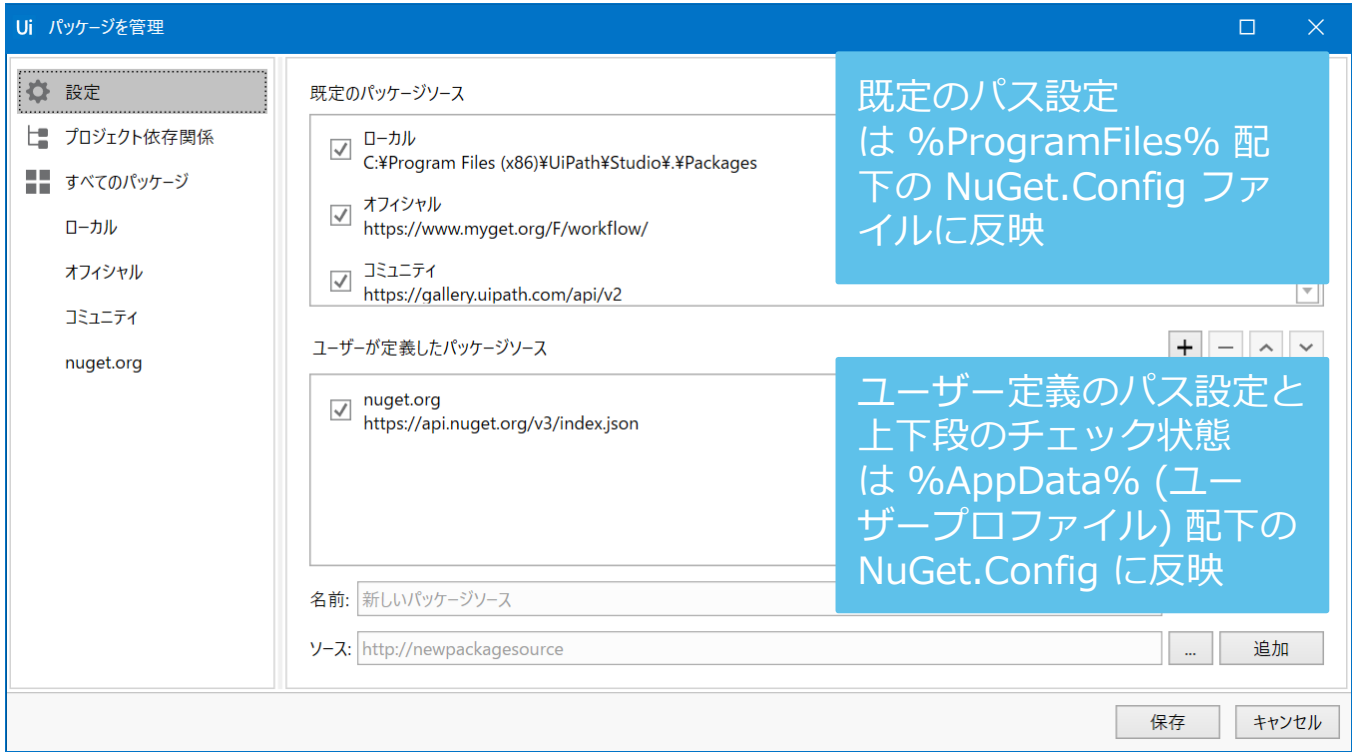
既存 WF は、実行時に引き続き Core アクティビティを参照します。新規 WF とマイグレーション済み WF は、分割後のパッケージを直接参照するようになります。



アクティビティの管理とフィード設計指針

Studio のパッケージソースを設計する

複数のパッケージソースを登録して利用可能です。



- 既定のパッケージソースを有効・無効の変更、ユーザー定義のパッケージソースの追加・削除が可能。変更び内容は NuGet.Config ファイルに反映される。
- NuGet.Config ファイルを直接変更する事で、既定のパッケージソースの項目を変更できる。

＊：外部インターネット接続がない環境では、リモートソースが構成されているとワークフローの起動が遅くなる場合があるため、Studio/Robot のインストール/バージョンアップ時にリモートソース設定を無効にすることを推奨。

現状では、任意のアクティビティ (インターネット上にある第三者が作成したものを含む) の配置・使用を禁止する手段はないことに注意が必要。

アクティビティのパッケージソースについて

アクティビティパッケージファイルは、パッケージソースに配置されます。
ローカルソースとリモートソースのパッケージファイルを配信する機能は同一ですが、ローカルソースの方がレスポンスが高速です。

#	パッケージソースの種別	分類	概要	例
1	ローカルパス	ローカルソース	ローカルPCのパス。ネットワークフォルダをローカルにマウントしたパスも使える	C:¥Program Files¥UiPath¥Studio¥Packages
2	UNC*1パス	リモートソース	ファイルサーバー上に共有したフォルダをソースとして利用できる	¥¥FileServer¥YourPackages
3	NuGet サーバーのURL	リモートソース	NuGet サーバーがホストするソース	https://api.nuget.org/v3/index.json
4	Orchestrator が内包する NuGet サーバ	リモートソース	Orchestrator は NuGet サーバーを内包しており、Orchestrator に接続した Studio/Robot では何らを設定することなく利用可能な状態となる	%installdir%¥UiPath¥uipath.settingsに自動で追加されるため、手動での設定は不要

*1：Universal Naming Conventionは、Windowsネットワーク上で共有されている様々な資源の位置を表記する標準的な記法です。


UiPath™
Thank you!