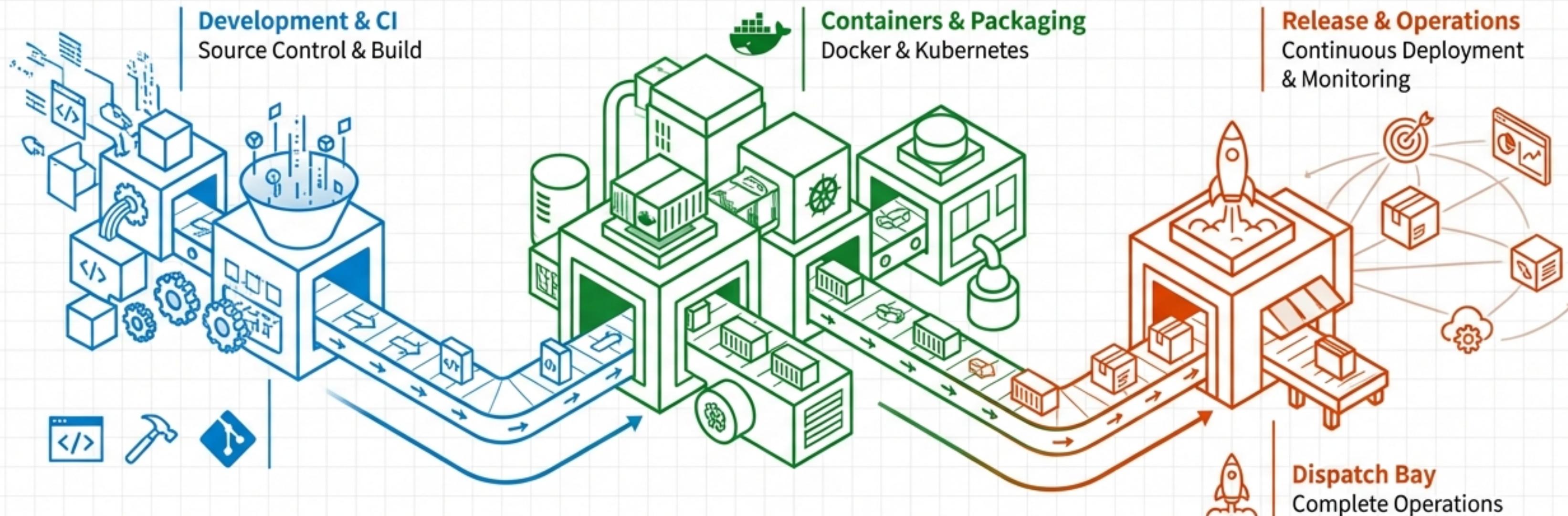


Azure DevOpsによる開発・運用基盤の全体像

高速かつ確実な「デジタル工場」の構築



本資料では、CI/CD、コンテナ化、リリース管理を統合した「デジタル工場」としての開発基盤戦略を提示します。
手作業を排した自動化ラインにより、開発チームのスキルセットに最適化した高速で確実なデリバリーを実現します。

本プロジェクトが目指す開発の姿



自動化によるスピード

手動作業を極小化。コードのコミットからビルドまで、Azure Pipelinesにより一連のプロセスを完全自動化し、開発リードタイムを短縮します。

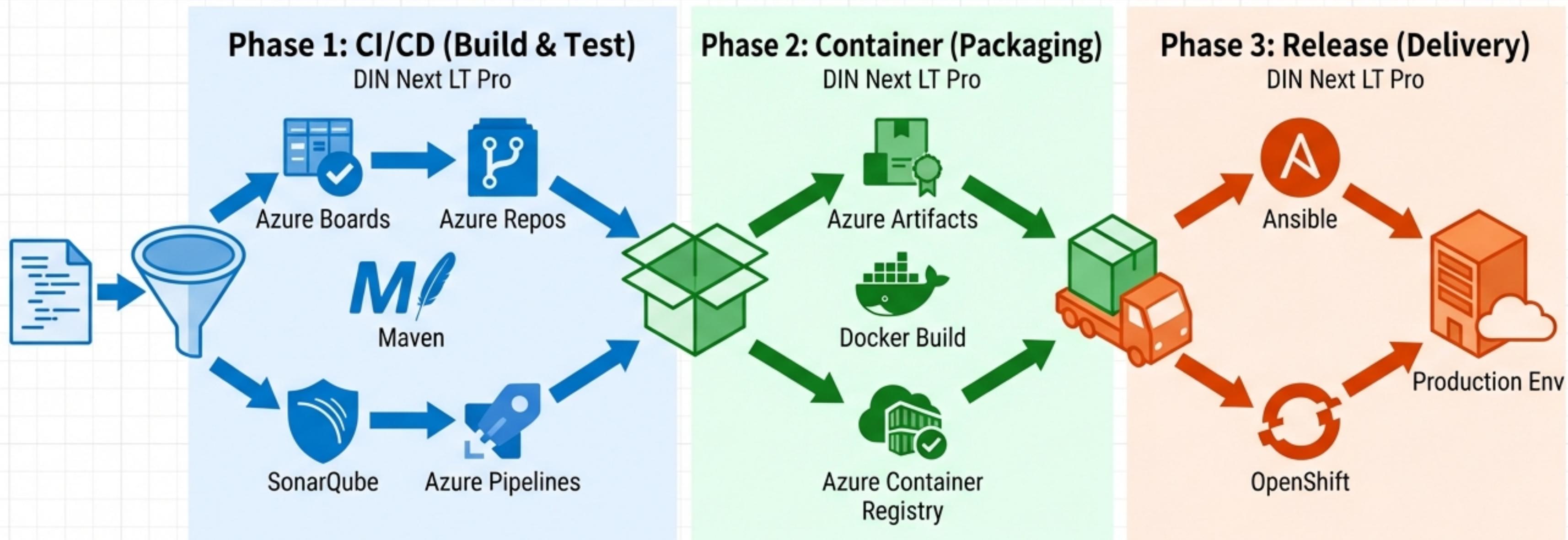
品質のゲートキーパー

SonarQubeとJacocoによる自動検査をラインに組み込み、欠陥のあるコードが後工程に流れるのを機械的に阻止します。

シンプルな運用

複雑なマージ作業やコンフリクト解消を回避するため、メインブランチを中心としたシンプルな開発フロー（トランクベース）を採用します。

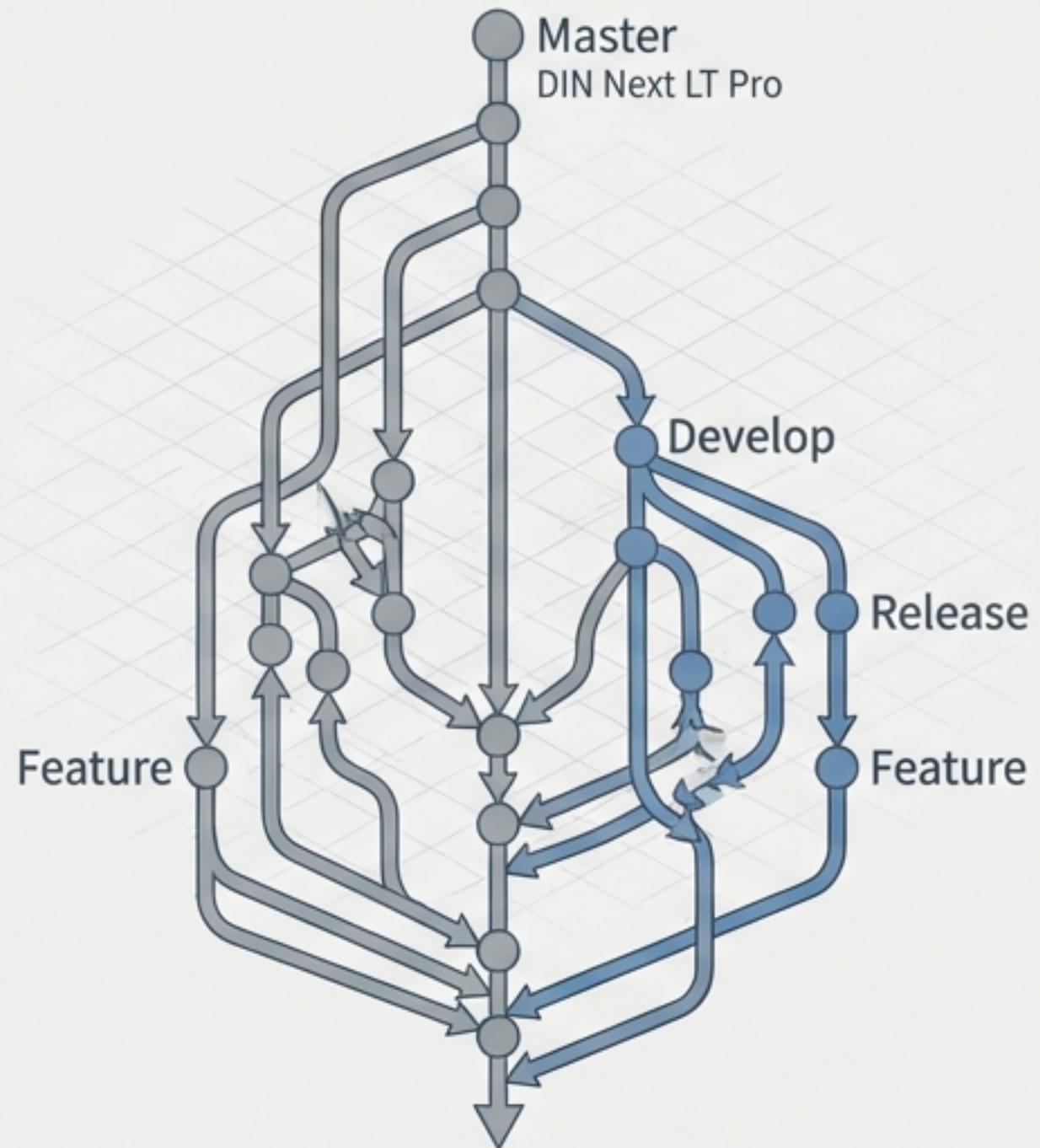
全体構成プロセスフロー



Noto Sans JP Bold (Build & Test) Noto Sans JP Regular, #E10y 5%, DIN Next LT Pro,
ソースコードから本番稼働まで、3つの自動化ステージを経てデリバリーされます。
DIN Next LT Pro

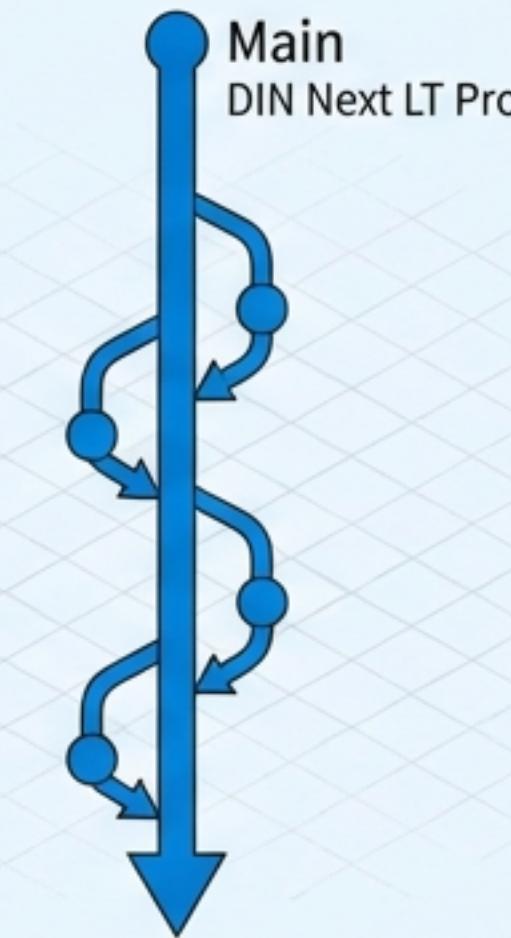
開発戦略：メインブランチ中心の開癝

パターンA：複雑なブランチ戦略 (Git Flow)



マージ作業が頻発し、コンフリクト解消のコストが高い。

パターンB：本プロジェクトの戦略 (Trunk-Based)

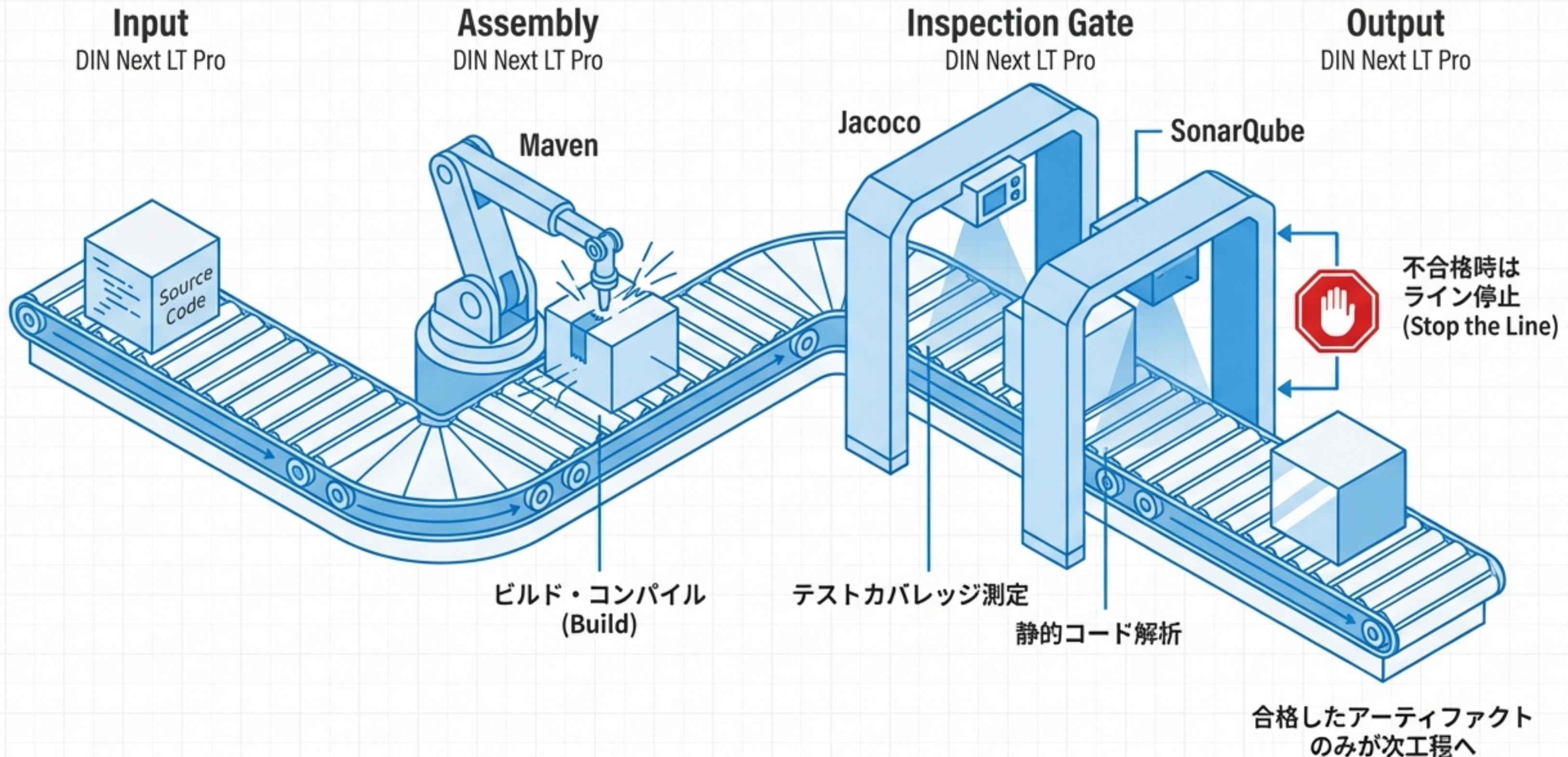


常に最新のコードを共有。マージ地獄を回避し、スピードを最優先。

主なメリット

- マージ作業の回避：複雑な競合解決の手間を削減
- 即時の統合：統合問題を早期発見
- 安全装置：Azure Pipelinesが自動テストで品質を担保

CIパイプライン：自動化された組立ライン



バージョン管理指針：スナップショットとリリース

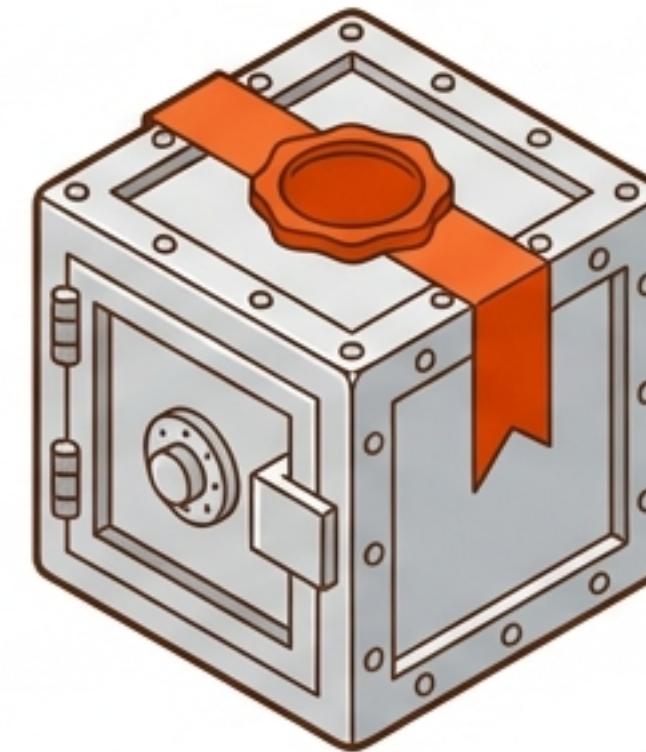
開発用：スナップショット (Snapshot)



ver 1.0.0-SNAPSHOT

日々上書きされる開発版。
常に最新の不安定な状態を共有する用途。

本番用：イミュータブル・リリース (Release)

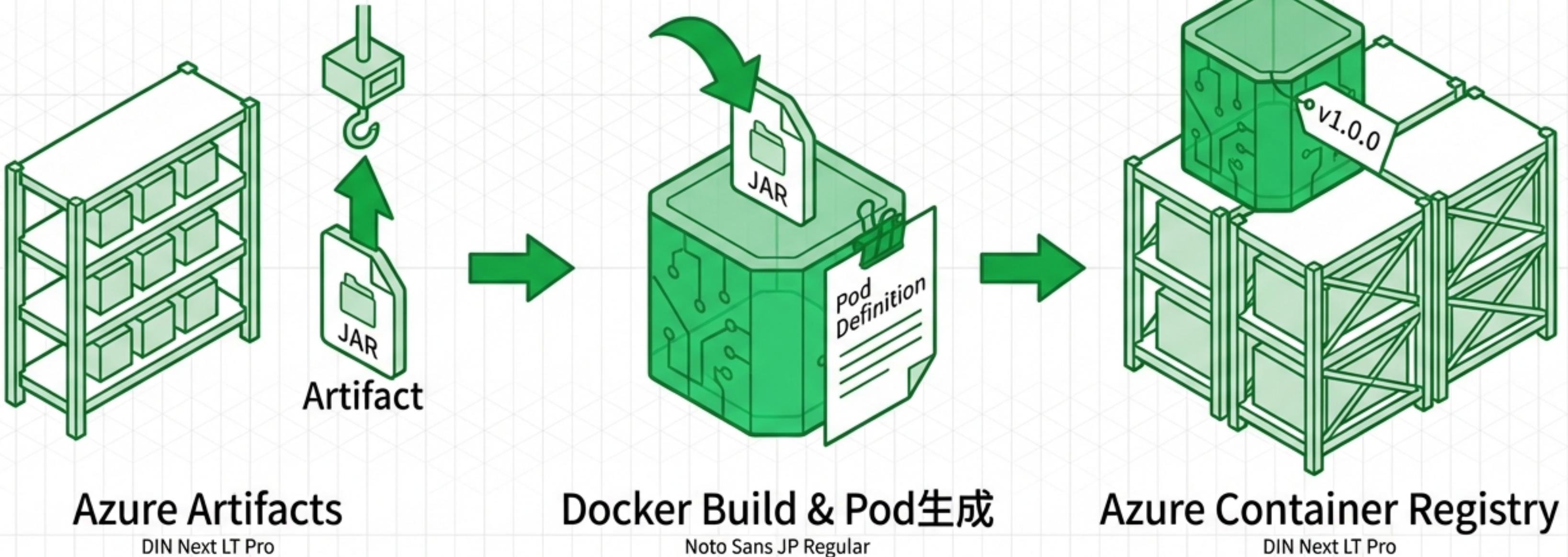


ver 1.0.0 (Immutable)

一度作成したら絶対に上書きされない不変のバージョン。
いつデプロイしても同じ結果を保証。

「不変のリリース」を作成することが、確実なロールバックの条件となります。

コンテナ生成と登録



Azure Artifacts

DIN Next LT Pro

Docker Build & Pod生成

Noto Sans JP Regular

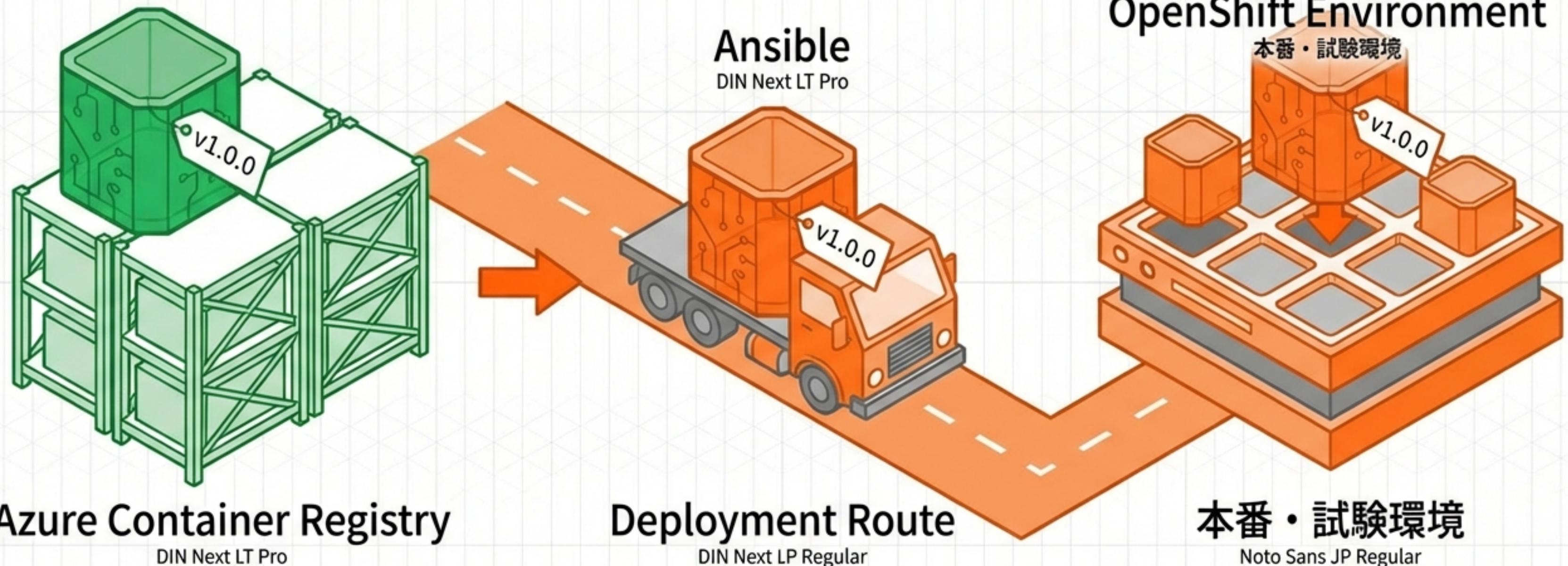
Azure Container Registry

DIN Next LT Pro

検証済みのアプリケーション (JAR) をランタイム環境ごとコンテナに封入し、
Azure Container Registry (倉庫) へ保管します。

Noto Sans JP Regular

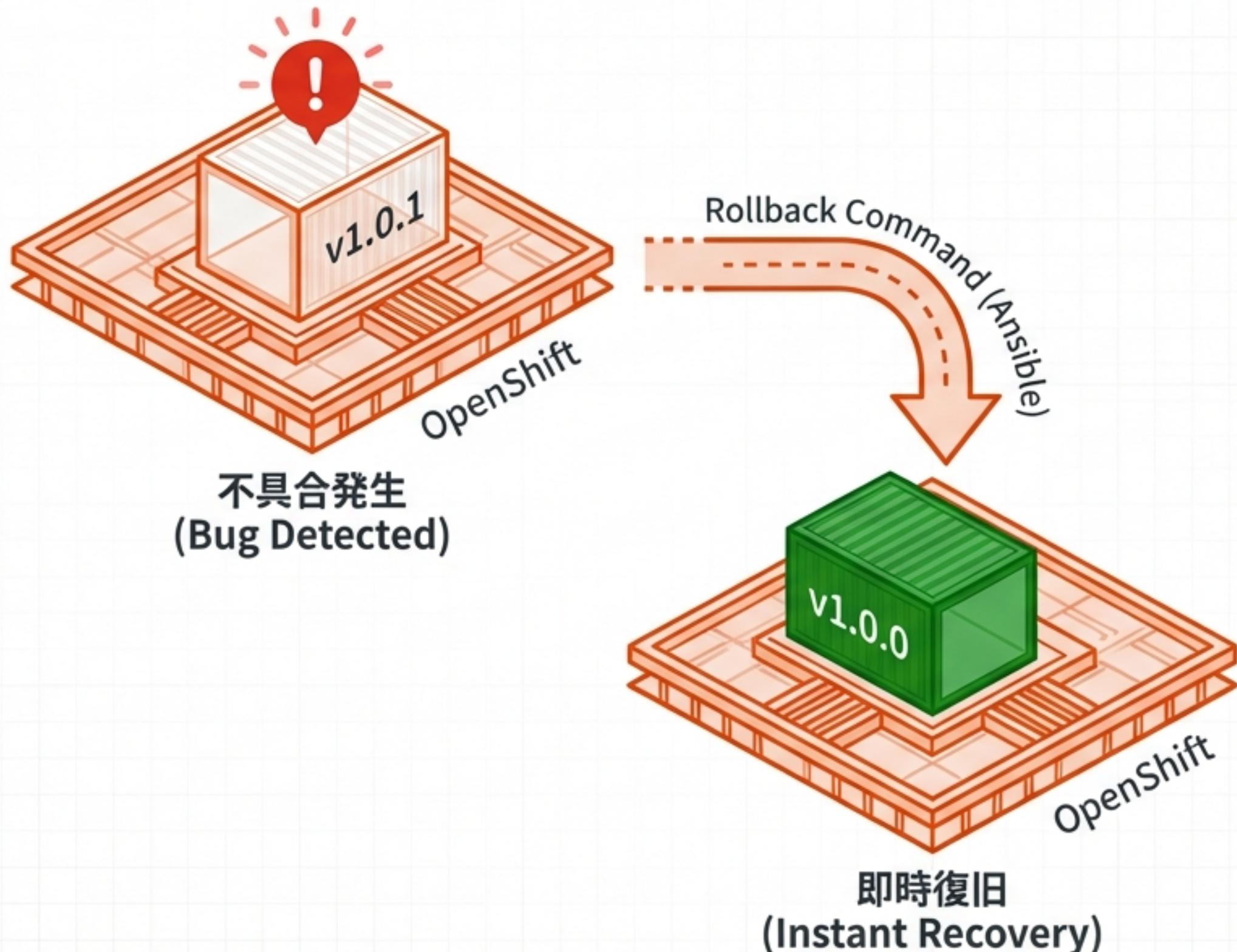
本番・試験環境へのリリース



Ansible（自動化ツール）がRegistryから指定されたバージョンのコンテナを引き出し、OpenShift環境へ確実に配置します。

Noto Sans JP Regular

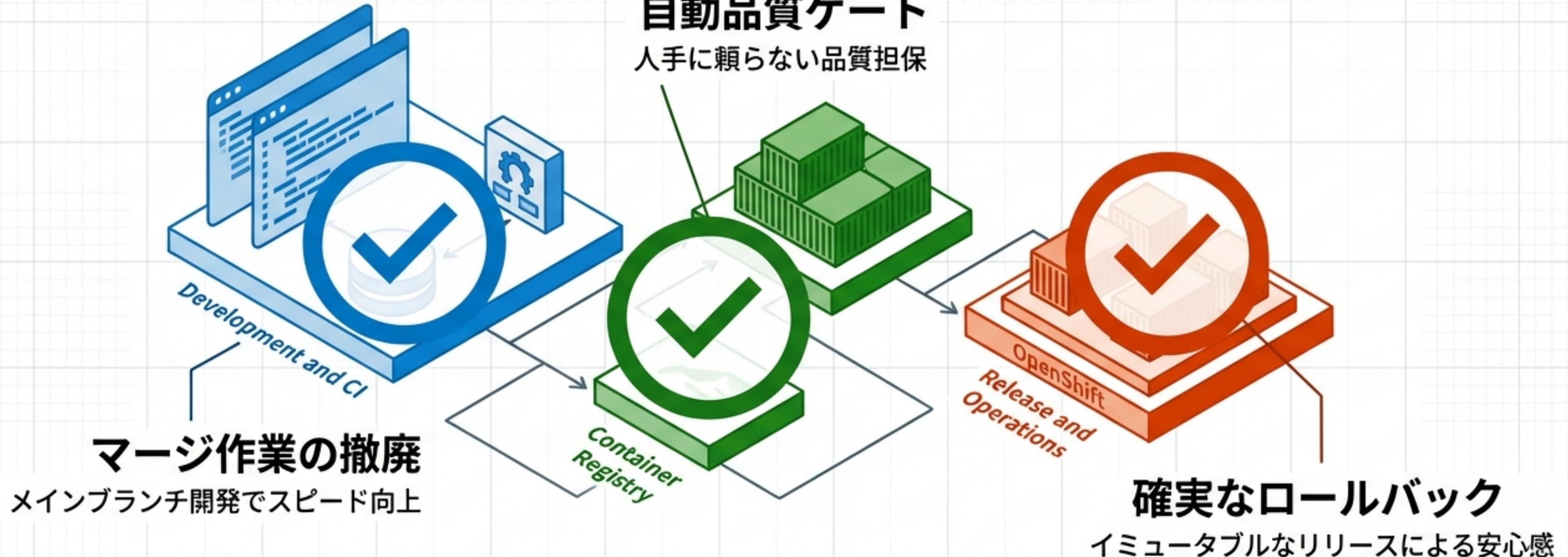
可用性とロールバック：幂等性の担保



幂等性 (Idempotency)

「何度実行しても同じ結果になる」特性。イミュータブルなリリース版を使用することで、修正パッチを当てるのではなく、正常な状態へ「時間を巻戻す」ような確実な復旧を実現します。

まとめ：本アーキテクチャの導入効果



開発スピードと運用安定性を両立する「デジタル工場」を実現します。