**基本ルール：システム運用**

**1. リソース管理**

* **CPU使用率:** 80%を警告閾値、95%を危険閾値として監視する。
* **メモリ使用率:** 75%を警告閾値、90%を危険閾値として監視する。
* **ストレージ:** 残り20%を切った場合に警告、10%を切った場合に危険アラートを発する。
* 警告閾値ではFactoryOverviewにアラート表示し、危険閾値では管理者への通知も実施する。

**2. エラー処理**

* **レベル1（軽微）：** 設定ファイルのリロード、キャッシュクリア、サービス再起動などを自動的に試行する。
* **レベル2（中程度）：** 依存サービスへの再接続、データ整合性チェック、バックアップからの復元などを自動的に試行する。
* **レベル3（重大）：** 修復不可と判断した場合は、即時に上位システムへ報告する。
* **報告形式:**
  + Factory Overviewでの視覚的アラート（色分け、アイコン）。
  + Chat with teamでの構造化エラーメッセージ（エラーコード、概要、影響、推奨アクション）。
  + 重大エラー時の詳細レポート自動生成（エラー発生時の環境変数、ログトレース、影響範囲分析）。

**3. セキュリティ**

* **暗号化:** 個人情報はAES-256により暗号化（保存時）、通信時はTLS 1.3を使用する。
* **アクセス制御:** 役割ベース（RBAC）でのアクセス管理を行い、最小権限の原則を適用する。
* **認証:** 多要素認証を導入し、セッション管理を厳格化する。
* **役割分担:**
  + **AIエージェント：** 異常検知（アクセスパターン、データ流出パターン等）、ログ記録、初期分析を行う。
  + **セキュリティ部門：** アラート検証、対応策決定、インシデント管理、セキュリティポリシー策定を行う。
* **連携フロー:** AIが異常を検知した場合、セキュリティチャネルに暗号化して報告し、セキュリティ部門が検証後、対応を実施する。

**4. システム監視**

* **アセンブリラインステータスの監視:**
  + **Running：** アクティブなタスク処理中、CPU/メモリ使用率が閾値以上の場合。
  + **Idle：** タスク待機中、CPU/メモリ使用率が閾値以下の場合。
  + **Maintenance：** 定期メンテナンス中または管理者による手動メンテナンス設定時。
* **追加監視指標:**
  + **API応答時間：** 200ms以上で警告、500ms以上で危険と判断する。
  + **タスク処理遅延：** 予定処理時間の120%超過で警告、150%超過で危険と判断する。
  + **エラー発生頻度：** 10分間に3回以上の同種エラーで警告、5回以上で危険と判断する。
  + **リソース使用効率：** CPU時間/処理タスク比率を監視する。
* **通知機能:**
  + 優先度ベースのアラート（緊急、警告、通知）を提供する。
  + 通知方法を多様化する（ダッシュボード、チャット、メール、モバイル通知）。
  + ユーザーが設定可能なアラート閾値とルールを提供する。

**5. ログ記録**

* **ログ情報の詳細度:**
  + **エラーログ：** エラーコード、メッセージ、発生時刻、関連コンポーネント、発生時の環境変数、コールスタックを記録する。
  + **操作ログ：** 操作者、操作内容、操作時刻、対象リソース、操作結果を記録する。
  + **システムログ：** リソース使用状況、起動/停止イベント、設定変更履歴を記録する。
  + **セキュリティログ：** 認証試行、権限変更、セキュリティイベントを記録する。
* **ログ保管とアクセス権限:**
  + **保管場所：** 初期はローカル暗号化ストレージ、将来的には専用のログサーバーに保管する。
  + **保管期間：** 操作ログ1年、エラーログ6ヶ月、システムログ3ヶ月（重要度によって延長可）。
  + **アクセス権限：** 管理者はフル権限、開発者は読取権限、AIエージェントは自身の関連ログのみ読取可とする。

**6. メンテナンス**

* **メンテナンス時間の柔軟性と制約:**
  + 最低メンテナンス頻度を月1回必須とし、2週間以上スキップ時に警告する。
  + 最長メンテナンス間隔を4週間とする（セキュリティ更新等の緊急度による例外あり）。
  + メンテナンス時間帯は利用頻度の低い時間帯を推奨し（デフォルト：土曜0時）、ユーザーは変更可能とするが、頻度は最低要件を満たす必要がある。
* **自動アップデートの許可範囲:**
  + マイナーバグ修正、セキュリティパッチ、パフォーマンス最適化を許可する。
  + 機能追加、UIの変更、ワークフロー変更、データ構造変更については人間の承認を必須とする。
* **自動アップデート条件:** システム負荷低下時、フォールバック計画が存在する場合、検証環境でのテスト済みであることを条件とする。

**7. データの整合性**

* **データ重複防止の仕組み:**
  + UUID/GUIDベースのタスクID、エージェントIDなどのユニーク識別子を採用する。
  + 関連データの適切な分割と参照関係の確立によりデータ正規化を行う。
  + 新規データ挿入前に既存エントリとの照合を行う。
  + 定期的な重複スキャンと統合プロセスを実施する。
* **矛盾検出方法:**
  + 事前定義された業務ルールに基づく整合性チェックを行う。
  + 連続データポイントの異常を検出する時系列分析を行う。
  + 複数ソースからのデータを照合する。
* **矛盾検出時の対応:** 低確信度の場合はフラグ付けを行い人間によるレビューを実施し、高確信度の矛盾の場合は自動修正ルールを適用する。

**8. UIに関連する運用**

* **UI情報の最新化方法:**
  + 重要指標は5-10秒間隔でリアルタイムモニタリングにより更新する。
  + 負荷の少ない状態情報は30秒-1分間隔で定期的ポーリングにより更新する。
  + 状態変化時には即時プッシュ通知により更新する。
* **アセンブリラインステータス更新トリガー:**
  + Running→Idle：タスクキューが空になった時、処理完了後の待機状態に入った時。
  + Idle→Running：新規タスク受信時、処理開始時。
  + ⇔Maintenance：スケジュールメンテナンス開始/終了時、管理者による手動切替時。
* **AIエージェントのUI操作権限:**
  + 状態情報の更新、動作ログの表示、軽微な設定調整を許可する。
  + システム再起動、構成変更、リソース割当変更は制限し、承認を必要とする。
* すべての操作について操作者（AIエージェント/人間）、操作内容、時刻を記録する。

**9. ディザスタリカバリ**

* **バックアップ戦略:** 重要データの日次増分バックアップ、週次フルバックアップを実施する。
* **リカバリポイント目標（RPO）：** 24時間以内（重要データは4時間以内）とする。
* **リカバリタイム目標（RTO）：** クリティカル機能は2時間以内、全機能は24時間以内とする。
* 主要コンポーネントの冗長化を行い、自動切替設定によるフェイルオーバー機構を設ける。