# 特許庁アーキテクチャ標準仕様書
# （別冊4）システム機能共通編

第1.1版

平成28年6月

特許庁

## 改訂履歴

| 項番 | 版数 | 作成日/改訂日 | 変更箇所 | 変更内容 |
|---|---|---|---|---|
| 1 | 1.1 | 平成28年5月31日 | 新規 | 章構成の変更，分冊化に伴い新規作成。 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# はじめに

## (1) 本書の位置づけ

本書は,『特許庁アーキテクチャ標準仕様書』の各要素における個別ルールのうち，特許庁システムの機能に関する共通的な内容を規定し，別冊として定めたものである。

本書で扱う内容は，特許庁内の標準・規約類文書で既に定められているものが多い。特許庁内の標準・規約類文書で定められているものは，原則それに従うこととするが，本書ではToBeアーキテクチャとの関係や，アーキテクチャの観点で補足すべき事項について記載する。本書で定めるアーキテクチャに関係する標準・規約類については,『特許庁アーキテクチャ標準仕様書』の本冊（以下,『本冊』と呼ぶ）の「表 (1)-1 本書に関係する標準・規約類の概要」を参照のこと。

## (2) 本書の利用者及び利用目的

本書は，個別システム刷新に関係するステークホルダ（情報技術統括室職員，特許庁PMO，システム利用者，原課，要件整理補助（支援）業者，調達支援業者，設計・開発ベンダ，システムインテグレーションベンダ等）向けに作成されたものであり，当該ステークホルダが本書を利用しシステムの構造を標準化するためのルールに従い個別システム刷新を行うことにより，段階的刷新を通じ特許庁システム全体として統一された保守性と移植性の高いシステムを実現することを目的とする。

## (3) 本書の文書構成

本書は，以下の章から構成される。

1章 システム機能に関する共通的なルール

システム機能（セキュリティやバックアップ等）に関する設計開発時に遵守すべきルールを定める。

本書においても『本冊』のルールの考え方[1]に基づき，分類されるルールを規定する。詳細は,『本冊』の「2. ルールの考え方」を参照のこと。

本書におけるルールの表記方法は，以下のとおり。

- ルールの前段に「目的」（ルールの目的）を記載することにより，ルールを遵守することで達成したい事柄を明確化する。
- 規約及び設計指針は連番を付与し列挙する。また，設計指針はルールの表記上，「指針」と記載する。
- 推奨や特例ルールも連番を付与し，付随する規約又は設計指針のルールの直後に字下げして記載する。

以下にルールの表記例を示す。

なお，表記例における「スコープ」の詳細については『本冊』の「2.1 スコープ（ルールの適用範囲）」を，「規約・指針・推奨・特例」の詳細については『本冊』の「2.2 強制力（ルールの裁量）」を参照のこと。

[1] 設計方針に基づいて段階的に刷新される各刷新対象システムの構築に必要となる，設計に関与するステークホルダ（設計・開発ベンダ等）の遵守事項（ルール）を，以下の観点で整理する。

- スコープ（ルールの適用範囲（システム））
- 強制力（ルールに含まれる設計に関与するステークホルダ（設計・開発ベンダ等）の裁量の範囲であり，設計指針，規約，推奨及び特例の4つに分類）

| |
|---|
| 目的:　XXXXXXXXX<br>スコープ:　XXXXXXXXX<br>規約1:　XXXXXXXXX<br>規約2:　XXXXXXXXX<br>　特例1:　XXXXXXXXX<br>指針1:　XXXXXXXXX<br>　特例1:　XXXXXXXXX<br>　特例2:　XXXXXXXXX<br>　推奨1:　XXXXXXXXX |

## (4) 本書の利用方法

本書の利用者及び利用方法について以下に示す。

表 (4)-1 本書の利用者及び利用方法

(○:利用する, ー:利用しない)

| 利用者 / 利用方法 | 情報技術統括室 | 特許庁PMO | システム利用者, 原課 | 要件整理補助業者, 調達支援業者 | 設計・開発ベンダ | システムインテグレーションベンダ | ハードウェアベンダ | オペレーションベンダ |
|---|---|---|---|---|---|---|---|---|
| システム構造の定型化(ルールの理解・遵守) | ○ | ○ | ○※ | ○ | ○ | ○ | ー | ー |
| 技術的整合性確保(コントロール及びチェック) | ○ | ○ | ー | ○ | ○ | ー | ー | ー |

※ルールに従い画面設計等の設計レビューに関与するために必要。

本書は詳細なルールを記載しているため, 必要な箇所をその都度参照してルールを確認するといった利用方法を想定している。

## (5) 本書の運用方法

本書の運用方法について以下に示す。

① 運用開始時期

平成28年6月から運用を開始する。

② 改定時期

平成29年3月末, 平成30年3月末及び平成31年3月末の3回の時期において改定を予定している。

③ 整備及び管理

『特許庁PMO標準・規約類における整備及び管理方針』に従う。

# － 目 次 －

# 1. システム機能に関する共通的なルール

本章では，システム機能（セキュリティやバックアップ等）に関する設計開発時に遵守すべきルールを定める。
本章にルールを記載するシステム機能の方式を以下に示す。

(1) セキュリティ方式
(2) 冗長化処理方式
(3) システム運用管理方式
(4) システム監視方式
(5) バックアップ方式
(6) リリース管理方式
(7) ログ管理方式

特許庁内にはシステム機能に関するガイドライン等が既にあるため，原則，ガイドライン等に従うものとするが，To Beアーキテクチャとの関係や，補足事項等を本章に記載する。

本章で参照するガイドライン等を以下に示す。

- 『ポリシー実施手順（開発編別紙「設計基準」）』
- 『LDAPアクセス運用』
- 『運用管理エージェント登録ガイドライン』
- 『バックアップ設計指針』
- 『特許庁システム リリースポリシー』
- 『パッチ適用方針』

## 1.1 セキュリティ方式設計のルール

| | |
|---|---|
| 目的: | 情報システムの構築について特許庁情報セキュリティポリシーに基づき実施すべき事項や判断基準を定めることにより，特許庁の情報セキュリティを確保する。また，サブシステム間での認識齟齬を防止する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | セキュリティ方式設計は，『ポリシー実施手順（開発編別紙「設計基準」）』を遵守して設計すること。また，「表 1.1-1『ポリシー実施手順（開発編別紙「設計基準」）』とToBeアーキテクチャの関係」に示す事項は，同表の「ToBeアーキテクチャとの関係」に従い設計すること。 |
| 指針2: | 仮想化ソフトのセキュリティは，「(3)仮想化ソフトのセキュリティ」に従い設計すること。 |

(1)　『ポリシー実施手順（開発編別紙「設計基準」）』への対応

セキュリティ方式設計は，『ポリシー実施手順（開発編別紙「設計基準」）』に準ずるものとする。

『ポリシー実施手順（開発編別紙「設計基準」）』の記載内容のうち，ToBeアーキテクチャとの関係を下表に示す。

表 1.1-1『ポリシー実施手順（開発編別紙「設計基準」）』とToBeアーキテクチャの関係

| 章節 | タイトル | 概要 | ToBeアーキテクチャとの関係 |
|---|---|---|---|
| 2.1 | 本人確認機能 | 全ての情報システムについて，本人確認を行う必要性の有無を検討すること。 | 「システム利用者（庁職員等）」の識別情報を利用し認証・認可を行う。<br><br>詳細は，「(2)D.認証・認可の処理方法」を参照のこと。 |
| 2.2 | アクセス制御機能 | 全ての情報システムについて，情報システム及びそれに保存されている情報へのアクセス制御を行う必要性の有無を検討すること。 | 他サブシステムの個別データベースにアクセスするには他サブシステムが提供するサービスを経由してアクセスする。<br>共通的に利用するデータベースにアクセスするには，DBアクセス基盤サービス又は配付される基盤APIからアクセスする。<br><br>詳細は，『本冊』の「3.3.3.4 個別データベースの構成及びアクセスルール」及び「3.3.5.1 共有データベースの構成及びアクセスルール」を参照のこと。 |
| 2.5 | 権限管理機能 | 全ての情報システムについて，権限管理を行う必要性の有無を検討すること。 | 「システム利用者（庁職員等）」や「システム」毎に，アクセス可能なシステム構成要素を制限する。<br><br>詳細は「(2)E.認証・認可を実施するシステム構成要素間のアクセス」を参照のこと。 |
| 2.6 | 共有識別子（グループID）の利用 | 権限管理を行う必要があると認めた情報システムにおいて，共有識別子（ID等）の利用許可については，情報システム毎にその必要性を判断すること。 | サブシステム毎に，アクセス可能なシステム構成要素を制限する。また，「システム」の識別情報を利用し認証・認可を行う。<br><br>詳細は，「(2)D.認証・認可の処理方法」を参照のこと。 |
| 2.9 | 証跡管理 | 全ての情報システムについて，証跡管理を行う必要性の有無を検討すること。 | ToBeアーキテクチャで導入されるプログラムプロダクトも，証跡管理のため，ログを出力するように設定する。<br>また，不正アクセス等のセキュリティ監査のため，ログを出力する。<br><br>詳細は「1.7 ログ管理方式設計のルール」を参照のこと。 |

| 章節 | タイトル | 概要 | ToBeアーキテクチャとの関係 |
|---|---|---|---|
| 2.10 | 監視機能 | 情報システムについて，情報セキュリティの侵害又はそのおそれのある事象の発生を監視する必要性の有無を検討し，必要があると認めた場合には，監視のために必要な措置を定めること。 | ToBeアーキテクチャで導入されるプログラムプロダクトも，監視の対象とする。<br><br>詳細は「1.4 システム監視方式設計のルール」を参照のこと。 |
| 2.18 | 入出力データの妥当性確認機能 | 開発するソフトウェアにおいて処理するデータ及び入出力されるデータの情報セキュリティに関する妥当性を確認する機能の必要性の有無を検討し，必要と認めたときは，その方法を適切に（例えば，HTMLタグやスクリプト等として機能する不正な文字列や通信過程において生じたデータ誤り等，データ処理の障害になる情報がデータ内に含まれない状態であること等）設計しなければならない。 | ToBeアーキテクチャにおいても同様に入出力データの妥当性の確認を行う。<br>具体的には以下のようなものがある。<br>● 入力データの長さや内容を検査し，無害化する機能を設けること。<br>● OSの関数，SQLコマンド等の呼び出しといった出力情報に不正なデータの混入を排除すること。<br>● 製品名及びそのバージョン，登録されているユーザID等，攻撃の糸口となり得る不必要な情報は出力しないこと。<br>● なりすましによるアクセスを防止するため，適切なセッション管理を行うこと。 |

(2)　認証・認可

A.　目的

認証・認可の目的を以下に示す。

認証:本人性を確認し，なりすまし等を防止する。
認可:サービスの利用やリソースへのアクセス等に対する権限を制御し，権限を持たない者からのアクセスを防ぐ。

B.　対象

認証・認可の対象を下表に示す。

表 1.1-2 認証・認可の対象

| 項番 | 対象 | 目的 |
|---|---|---|
| 1 | システム利用者（庁職員等） | 権限を持たない利用者からのアクセスを防ぐ。<br>認証・認可は，原則，『LDAPアクセス運用』に従うものとする。 |
| 2 | システム | 特許庁のシステム全体で共通的に利用可能な機能が，意図していないシステムからアクセスされることを防ぐ。 |

C.　識別情報の管理方法

認証・認可のための識別情報の管理方法を下表に示す。

表 1.1-3 識別情報の管理方法

| 項番 | 対象 | 識別情報の管理場所 | 管理する情報 | 取得方法 |
|---|---|---|---|---|
| 1 | システム利用者（庁職員等） | リソース管理サブシステム（既存システムでは，共通テーブル管理システム） | 利用者，部署，権限等 | LDAPより取得する。 |
| 2 | システム | 認証・認可する側のシステム内 | システムの認証・認可情報 | - |

D.　認証・認可の処理方法

認証・認可の処理方法を下表に示す。

表 1.1-4 認証・認可の処理方法

| 項番 | 対象 | 認証・認可される側 | 認証・認可する側 |
|---|---|---|---|
| 1 | システム利用者（庁職員等） | ①　職員等の識別情報(IDやパスワード等)を送信する。利便性向上のため，操作端末上でシングルサインオン(SSO)を実現するための認証基盤の利用を推奨する。 | ②　認証対象の識別情報を用い，リソース管理サブシステム（既存システムでは，共通テーブル管理システム）から職員等の情報（利用者，部署，権限等）をLDAPより取得することにより認証を行う（認証）。<br>③　取得した職員等の情報を用いてアクセス権限の制御を行う（認可）。 |
| 2 | システム | ①　自システムの識別情報(IDやパスワード等)を送信する。 | ②　認証対象のシステムの識別情報が正しいか照合し認証を行う（認証）。<br>③　認証と同様に，予め管理されている認可情報と照合し，アクセス権限の制御を行う（認可）。 |

BPMS以外のプログラムプロダクトの認証・認可の処理方法は，「システム」での認証・認可になる。
BPMSに対する認証・認可の処理方法を以下に示す。

(A) BPMSに対する認証認可の処理方法

BPMSに対する認証認可の対象は，以下とする。

- 認証認可の対象に製品の制約がない場合は，BPMSに対して「システム」での認証・認可とする。
- 「システム利用者（庁職員等）」による認証認可を必要とする製品の場合，「システム利用者（庁職員等）」での認証を行う。

例えば，BPMSに渡された「システム利用者（庁職員等）」の識別情報を使用して，BPMSがLDAPサーバで認証を行う場合等がある。

BPMSがLDAPサーバから「システム利用者（庁職員等）」の情報を取得するために「システム」での認証を必要とする場合がある。この場合は，BPMSからLDAPサーバに対しても「システム」での認証を許容する。このようなケースにおける認証・認可の手順の例を以下に示す。

① BPMSへの認証・認可は「システム」で行い，「システム利用者（庁職員等）」の識別情報はサービスの入力項目として渡す。ここでは「システム利用者（庁職員等）」自身に対する認証・認可を行うのではなく，役割に関する情報を取得することが目的であるため「システム利用者（庁職員等）」のIDを渡すだけでよく，パスワードを渡す必要はない。

② 次に，BPMSはLDAPサーバにアクセスし（ここで「システム」での認証を行う），システム利用者（庁職員等）の役割に関する情報を取得し，「システム利用者（庁職員等）」の役割に応じたアクティビティの検索等に使用する。

E. 認証・認可を実施するシステム構成要素間のアクセス

認証・認可を実施するシステム構成要素間のアクセスを下表に示す。

表 1.1-5 システム構成要素間のアクセス

| 項番 | 対象 | 識別情報送信元システム構成要素 | 認証する側のシステム構成要素 |
|---|---|---|---|
| 1 | システム利用者（庁職員等） | ● ブラウザ<br>● リッチクライアント | ● プレゼンテーションロジック |
| 2 | | ● ブラウザ<br>● プレゼンテーションロジック[2] | ● BPMS[3] |
| 3 | システム | ● BPMS<br>● BPMS補完機能<br>● プレゼンテーションロジック<br>● 業務アプリケーション（ユーザ）<br>● 業務アプリケーション（システム）<br>● 業務アプリケーション（バッチ）<br>● ESB | ● 業務アプリケーション（システム） |
| 4 | | ● プレゼンテーションロジック[4]<br>● BPMS<br>● BPMS補完機能<br>● ESB<br>● 業務アプリケーション（システム）[5]<br>● 業務アプリケーション（バッチ）[5] | ● BPMS |

[2] 「D.(A)BPMSに対する認証認可の処理方法」に示す「「システム利用者（庁職員等）」での認証を必要とする製品の場合」のみに使用するアクセスである。

[3] ブラウザからBPMSへのアクセスは原則，行わない。ただし，製品が提供する機能（BAM等）をブラウザから直接利用するような場合は庁職員等の認証を行うものとする。

[4] 当該職員に割り当てられているタスクを取得する等の処理を実現する場合，「システム利用者（庁職員等）」の識別情報をワークフローAPIに指定することで実現するものとする。

[5] アクセスパスの特例において認証・認可を実施するアクセスを指す。アクセスパスの特例については，『本冊』の「3.1.1.3.1.2 システム構成要素間のアクセスパス」を参照のこと。

| 項番 | 対象 | 識別情報送信元システム構成要素 | 認証する側のシステム構成要素 |
|---|---|---|---|
| 5 | システム(前ページからの続き) | ● プレゼンテーションロジック<br>● BPMS<br>● ESB<br>● 業務アプリケーション(システム)[5]<br>● 業務アプリケーション(バッチ)[5] | ● BPMS補完機能 |
| 6 | | ● 業務アプリケーション(ユーザ)<br>● 業務アプリケーション(システム)<br>● 業務アプリケーション(バッチ) | ● BRMS |
| 7 | | ● BPMS<br>● BPMS補完機能<br>● 業務アプリケーション(ユーザ)<br>● 業務アプリケーション(システム)<br>● 業務アプリケーション(バッチ)<br>● 外部システム互換機能 | ● ESB |
| 8 | | ● 業務アプリケーション(ユーザ)<br>● 業務アプリケーション(システム)<br>● 業務アプリケーション(バッチ)<br>● ESB<br>● 外部システム互換機能 | ● DBアクセス基盤サービス[6] |
| 9 | | ● 業務アプリケーション(ユーザ)<br>● 業務アプリケーション(システム)<br>● 業務アプリケーション(バッチ) | ● 個別データベース |
| 10 | | ● 業務アプリケーション(ユーザ)<br>● 業務アプリケーション(システム)<br>● 業務アプリケーション(バッチ)<br>● DBアクセス基盤サービス<br>● 外部システム互換機能 | ● 共有データベース |

(3)　仮想化ソフトのセキュリティ[7]

仮想化技術を導入した場合は，攻撃者がなりすましを行い仮想化ソフトの特権権限を不正に取得した場合の全仮想マシンへの影響が懸念される。

そのため，仮想化ソフトについても他のプログラムプロダクトと同様にセキュリティ方式設計を行うこと。

[6] 基盤機能層は責務としてビジネスロジックを持たないため，業務に依存するシステム利用者を対象とした認証・認可ではなく，システムを対象とした認証・認可とする。基盤機能層の責務については『本冊』の「3.3.5 基盤機能層及びデータベース層のルール」を参照。

[7] 以降の章節では，仮想化における考慮が必要な場合のみ考慮点を記載しており，仮想化に関係するルールがないものについては特に記載しないものとしている。

## 1.2 冗長化処理方式設計のルール

| | |
|---|---|
| 目的: | 冗長化を実現するための物理構成を定型化し，物理構成レベルの変更の影響の具体的把握を容易にする。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 規約1: | クラスタ構成は，「(1)冗長化の方法（クラスタ構成）のルール」に従うこと。 |
| 特例1: | プログラムプロダクトの機能や構成の仕方によって，より費用対効果の高い構成が実現できる場合は，「(1)冗長化の方法（クラスタ構成）のルール」を逸脱することを許容する。 |
| 指針1: | 仮想化における冗長化処理は，「(3)仮想化に対する考慮」に示す事項を考慮して設計すること。 |
| 指針2: | セッションにおける冗長化処理は，「(4)セッションにおける考慮」に示す事項を考慮して設計すること。 |
| 推奨1: | セッションレプリケーション等によるセッション保護機能は利用しないことを推奨する。 |

本節ではSLAに基づき冗長化する場合に，遵守すべき技術的なルールについて記載する。

(1)　冗長化の方法（クラスタ構成）のルール

冗長化の方法（クラスタ構成）として，スケールアウトに適した論理ノードはアクティブ-アクティブ構成とし，スケールアウトに適さない論理ノードはアクティブ-スタンバイとする。冗長化の方法（クラスタ構成）と対象論理ノードを下表に示す。

表 1.2-1 冗長化の方法（クラスタ構成）と対象論理ノード

| 項番 | クラスタ構成 | 対象論理ノード | 補足説明 |
|---|---|---|---|
| 1 | アクティブ-アクティブ | スケールアウトに適した論理ノード | 負荷分散装置又は高可用性ソフトウェアを用いて冗長化する。<br>例えば，Webサーバ等。 |
| 2 | アクティブ-スタンバイ | スケールアウトに適さない論理ノード | 例えば，DBサーバ等。 |

論理ノード毎にスケールアウトに適しているか否かは，『本冊』の「図 3.1-17 ソフトウェア構成図」を参照のこと。

(2)　冗長化の方法(クラスタ構成)の例

A.　アクティブ-アクティブの例

Webサーバは，複数台のサーバをアクティブ状態にして負荷分散装置が使用するサーバを切り替えて使用する。障害発生時は，障害が発生したサーバ以外で業務を継続する。

アクティブ-アクティブのイメージを下図に示す。

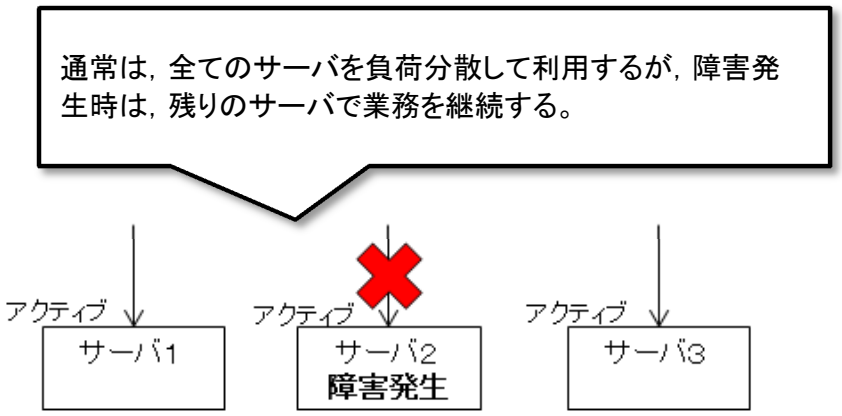


図 1.2-1 アクティブ-アクティブのイメージ

B.　アクティブ-スタンバイの例

DBサーバは，1台をアクティブ状態にして通常は，そちらを利用する。障害発生時は，スタンバイとなっているサーバをアクティブにして処理を行うよう切り替える。

アクティブ-スタンバイのイメージを下図に示す。

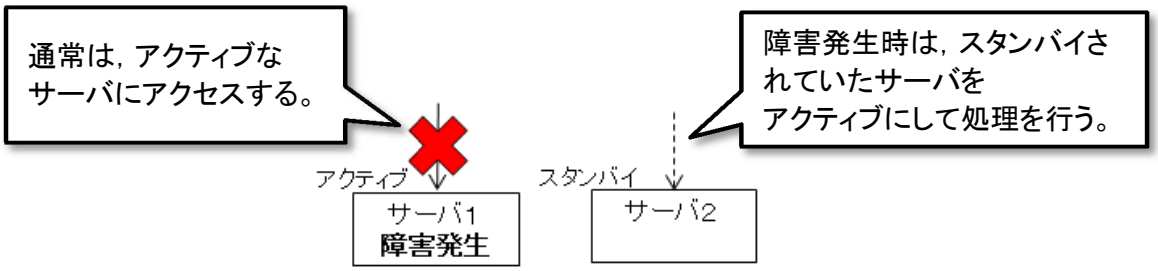


図 1.2-2 アクティブ-スタンバイのイメージ

(3)　仮想化に対する考慮

A.　仮想化対象外のサーバ

仮想化の対象外のサーバを以下に示す。以下のサーバは，仮想環境上に配置しないこと。

- ActiveDirectoryのドメインコントローラ
- 死活監視を行うサーバ（監視業務を行うサーバを仮想化し，その仮想マシンが動作しなくなった場合，障害の切り分けが困難となるため）
- NTPサーバ
- 仮想環境上での動作を保証していないソフトウェアを搭載するサーバ

B.　仮想化における冗長化技術について

仮想化技術のうち，冗長化に関わる技術を下表に示す。

表 1.2-2 仮想化における冗長化技術

| 項番 | 技術 | 概要 | 補足説明 |
|---|---|---|---|
| 1 | ライブマイグレーション | ある物理マシン上で稼動する仮想マシンを，停止させることなく丸ごと別の物理マシン上に移動させること。又はそのような機能。 | ● 仮想化ソフトが提供する。<br>● 通常，ハードウェアのメンテナンス時に使用する。<br>● 厳密には切り替え時に瞬断（製品にもよるが一般的にはミリ秒単位以上）が発生するが，セッションやメモリ情報を全て引き継ぐことにより，ユーザに使用中の仮想マシンが移動したことは分からないようにすることも可能。 |
| 2 | HAクラスタ | あるゲストOSを同一物理マシン上の別空間，別物理マシン上の仮想環境又は別物理マシン上の物理環境に移動させること若しくはそのような機能。 | ● クラスタリングソフトが提供する。<br>● 通常，障害発生時に使用する。<br>● セッションやメモリ情報の引継ぎは，使用するクラスタリングソフトの機能に依存する。 |

障害発生時，ライブマイグレーションだけでも業務継続は可能であるが，仮想マシンの数が多かったり信頼性に対する要件が特に高い仮想マシンが含まれる場合には切り替え時間が要件を満たさない可能性もあるため，HAクラスタと組み合わせて使用することも視野に入れ，サブシステム毎に検討するものとする。

(4)　セッションにおける考慮

画面系のオンライン処理では，複数ページにまたがる一連の処理を実現するため，UI層とプレゼンテーション層との間の通信においてセッションを利用する。通常，冗長化されたAPサーバのいずれか1つにセッションが管理される。

そのため，サーバをアクティブ-アクティブにしている場合，負荷分散装置で対象となるセッションを管理しているAPサーバへ振分けを行う。

ただし，セッションを管理しているサーバが故障した場合，セッションにアクセスできなくなり，故障したサーバのセッションを利用して業務を行っていたクライアントにはエラーが返却されることになる。対策としてはセッションレプリケーション機能等で，セッションをクラスタリングするか，セッション情報を別のサーバ上のRDBMSで管理する方法がある。

セッションレプリケーションのイメージを下図に示す。ただし，製品により実現方式は異なる。

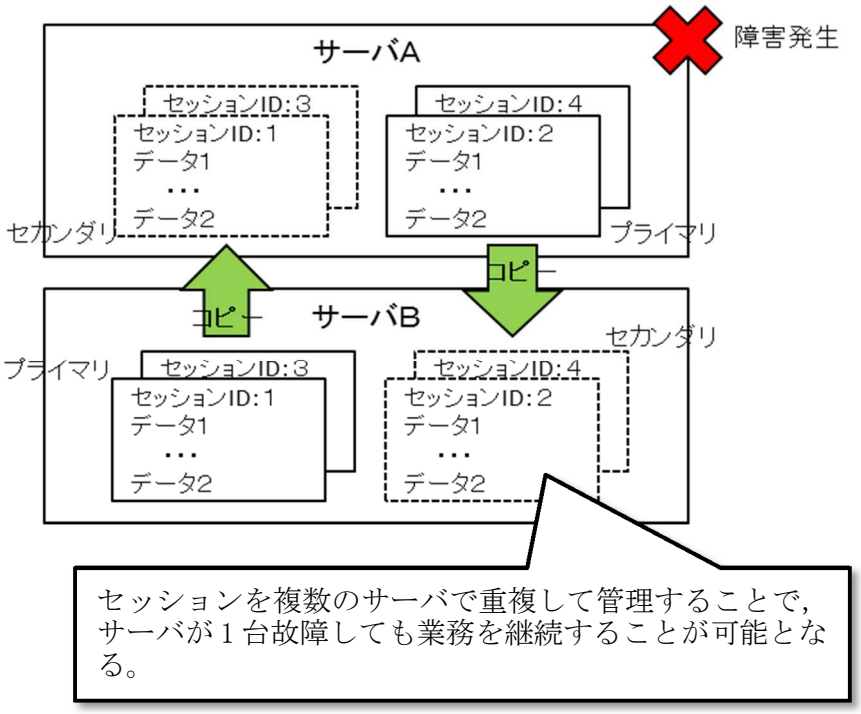


図 1.2-3 セッションレプリケーションのイメージ

しかし，これらの機能は，多くのアクセスが集中した場合に性能が劣化したり，レプリケーションの処理が新たな障害発生の原因になったりする場合がある。また，サーバの障害以外にも，Webブラウザのフリーズやユーザの誤った終了操作によってセッションが失われることもあるため，画面からあまり多くの情報を入力させないようにしたり，途中のページで固有DBに情報を保存し，後に再開可能とする等して，やり直しによる業務継続が可能であるような仕様にするべきである。

よって，ToBeアーキテクチャとしては，セッションレプリケーション等によるセッション保護機能は利用しないことを推奨する。

ただし，ユーザ利便性やSLAを遵守する上で，セッションレプリケーション機能等の導入が必要かサブシステム毎に検討するものとする。

## 1.3 システム運用管理方式設計のルール

| | |
|---|---|
| 目的: | ジョブ管理やバックアップ運用等の統合的なシステム運用管理を行うための指針に従うことにより，サブシステム間での認識齟齬を防止するほか，特許庁システムの統合的な運用を実現し，特許庁システムの安定稼動を確保する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | システム運用管理方式は，『運用管理エージェント登録ガイドライン』を遵守して設計すること。 |
| 指針2: | 業務閉塞は，「(1)B.業務閉塞のルール」に示すルールに従い設計すること。 |
| 推奨1: | オンライン運転時間だけでなく，システム的な運転時間も特許庁全体で極力，統一することを推奨する。 |
| 特例1: | やむを得ず，BPMS内部から経過時間指定での実行が必要な場合は，エラー終了状態になることを想定して開局時の処理を設計すること。 |

システム運用管理方式は，『運用管理エージェント登録ガイドライン』に準ずるものとする。

(1)　業務閉塞

ToBeアーキテクチャにおける業務閉塞に関わるルールを以下に示す。

A.　閉塞状態と運転時間

閉塞状態のステータスを下表に示す。

表 1.3-1 閉塞状態のステータス

| ステータス | 状態 |
|---|---|
| 開放 | クライアントからのリクエストを受け付ける状態 |
| 予閉塞 | クライアントからの新規リクエストを受け付けない状態 |
| 閉塞 | クライアントからのリクエストを受け付けず，仕掛かり中処理がない状態 |

ToBeアーキテクチャでは，他サブシステムに公開されるサービスがあるため，オンライン運転時間が過ぎ，画面からの操作が閉塞された後もBPMS上のシステムタスクが完了するまでサービスの提供が必要となる。このようにオンライン運転時間後も，システムが動作している時間帯を「システム的な運転時間」と呼ぶ。運転時間のイメージを下図に示す。

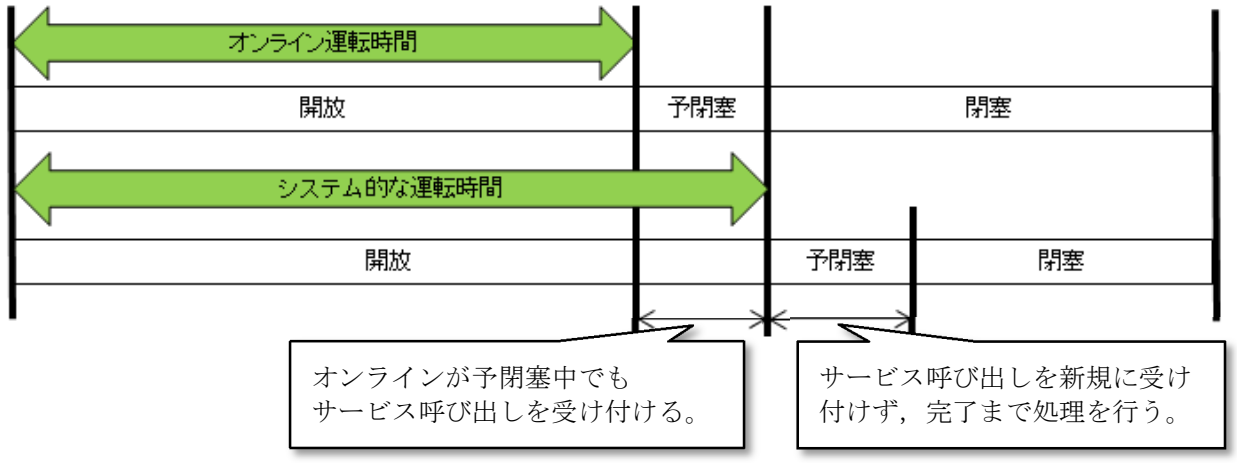


図 1.3-1 運転時間のイメージ

B.　業務閉塞のルール

業務閉塞のルールを以下に示す。

- 予閉塞から閉塞までの時間（仕掛かり中処理が正しく終了するまでの時間）については，サーバや業務処理によってリクエストの滞留量が異なるため，サブシステム毎に設計するものとする。ただし，上記を考慮しているにも関わらず，閉塞時に仕掛かり中処理がある場合には，全てのオンライン処理を強制的に終了し，閉塞状態に入るものとする。

- 閉塞状態の切り替えは自動的に行うため，ジョブ管理システムにスケジュール登録して自動で切り替えられるように設計すること。

- 他サブシステムに公開されるサービスを考慮し，オンライン運転時間だけでなく，システム的な運転時間も特許庁全体で極力，統一することを推奨する。

- BPMS内部から経過時間指定等での実行は行わないように設計すること（ビジネスプロセス上のタイマーイベントによる経過時間等の指定を行わないようにする。）。
  理由：BPMSでは，前回の実行からn時間後というように経過時間等を指定してBPMS内部からビジネスプロセスインスタンスへのアクセスをすることが可能である。このような設定をしている場合，オンライン処理が業務閉塞していてもBPMSのバックアップ等が正常に実施できない可能性があるためである。
  特例：やむを得ず，経過時間指定等での実行が必要な場合は，エラー終了状態になることを想定して開局時の処理を設計すること。

- BPMSの停止／再開に時間がかかる場合は，オンラインバックアップが可能な製品を採用すること。
  理由：特許庁システムでは，多くのビジネスプロセスインスタンスが起動されるため，ビジネスプロセスインスタンスの停止や再開には時間がかかると想定される。BPMSを停止してバックアップする場合，業務閉塞中にはバックアップが完了しない可能性があるからである。

## 1.4 システム監視方式設計のルール

| | |
|---|---|
| 目的: | KPIの変更，分析データの変更等の環境変化によるシステムの影響範囲を極小化する。また，BI，DWH，BAMのような業界標準的な技術を使用することにより，ベンダロックインを排除する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | KPIの測定，可視化，定期的分析は，「(2)稼動統計の監視　A」に従って設計すること。 |
| 推奨1: | KPIを測定，可視化し定期的にKPI達成の分析・評価を支援する仕組みは，BI（Business Intelligence），DWH（Data Warehouse）を使用して実現することを推奨する。 |
| 指針2: | KPIのリアルタイム確認は，「(2)稼動統計の監視　B」に従って設計すること。 |
| 推奨1: | 業務の実施状況をリアルタイムに確認し，ビジネスプロセス遂行上の問題発見や予防措置を支援する仕組みは，BPMS製品が持っているBAM（Business Activity Monitoring）を使用して実現することを推奨する。 |

| | |
|---|---|
| 目的: | 特許庁システムの統合的な運用を実現し，特許庁システムの安定稼動を確保する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | ハードウェアやソフトウェアの監視は，「(1)ハードウェアやソフトウェアの監視」に従って設計すること。 |
| 指針2: | SLA達成状況確認は，「(2)稼動統計の監視　C」に従って設計すること。 |

(1)　ハードウェアやソフトウェアの監視

ToBeアーキテクチャにおいても，『運用管理エージェント登録ガイドライン』に従い，システム監視を行うものとする。

ToBeアーキテクチャで導入されるプログラムプロダクト[8]も，監視の対象とする。

なお，画面操作でレスポンスが悪い場合やバッチが想定時間内に完了しない等，パフォーマンスに問題が見つかった場合，原因箇所を特定できるよう対処を行うこと。以下に例を示す。

- ログファイルに処理の開始日時と終了日時が分かるように処理名や日時を出力する。
- プログラムプロダクトにパフォーマンス等を確認する機能がある場合は，それを利用する。

(2)　稼動統計の監視

稼動統計の監視に関するルールを以下に示す。

A.　KPIの測定，可視化，定期的分析

- KPIを測定，可視化し定期的にKPI達成の分析・評価を支援する仕組みを構築する場合は，KPIの変更，分析データの変更等の環境変化に対して，容易に対応できる方式を採用すること。
  - BI（Business Intelligence），DWH（Data Warehouse）

B.　KPIのリアルタイム確認

- KPI達成のための更なる取り組みとして，日々の業務の中で，業務の実施状況をリアルタイムに確認し，ビジネスプロセス遂行上の問題発見や予防措置について迅速な対応を行うことで，ビジネスプロセスの効率を向上させることが有効である。このような活動を支援する仕組みを構築する場合は，KPIの変更，分析データの変更等の環境変化に対して，容易に対応できる方式を採用すること。また，以下のプログラムプロダクトを利用することを推奨する。
  - BPMS製品が持っている，BAM（Business Activity Monitoring）

C.　SLA達成状況確認

- SLAの達成状況を報告できるようにするため，応答時間を計測可能な箇所でログ出力し，機械的にSLA測定が可能な情報を蓄積するようにすること。ログについては，「1.7 ログ管理方式設計のルール」を参照のこと。

[8] 詳細は，『別冊3 プログラムプロダクト編』の「表 1.1-1 プログラムプロダクト一覧」を参照のこと。

## 1.5 バックアップ方式設計のルール

| | |
|---|---|
| 目的: | データリストア作業が必要となる障害が発生した場合のバックアップデータの取得について，リカバリポイントまで確実にデータ復旧を行えるようにする。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | バックアップ方式は，『バックアップ設計指針』に遵守して設計すること。 |
| 指針2: | BPMSのバックアップ方式は，オフラインバックアップにする場合，ビジネスプロセスインスタンスの停止及び再開に時間がかかることを考慮して設計すること。 |

ToBeアーキテクチャにおいても『バックアップ設計指針』に従い，バックアップを行うものとする。

ただし，BPMSのバックアップ方式をオフラインバックアップにする場合，ビジネスプロセスインスタンスの停止及び再開に時間がかかることを考慮すること。

## 1.6 リリース管理方式設計のルール

| | |
|---|---|
| 目的: | 本番環境へ正しい資材を確実に適用することにより, リリース後の特許庁システムの安定稼動を確保する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 規約1: | リリース管理方式は,『特許庁システム リリースポリシー』に従うこと。 |
| 指針1: | リリース時のパッチの適用は,『パッチ適用方針』に遵守して設計すること。 |

ToBeアーキテクチャにおいても, 以下に従いリリースやパッチ適用等が実現できるよう設計すること。

- 『特許庁システム リリースポリシー』
- 『パッチ適用方針』

## 1.7 ログ管理方式設計のルール

### 1.7.1 ログ出力の目的

| | |
|---|---|
| 目的: | ログを目的別に定義し遵守することにより，エラーや障害の対処，キャパシティ管理，監査，セキュリティ，SLA遵守確認，システム改善，デバッグといった目的の達成を効率的に行えるようにする。また，運用監視システムとの連携ルールを定めることにより，運用監視システムと運用監視の対象となるサブシステムとの間の認識齟齬を防止する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | OS，プログラムプロダクト，アプリケーションが出力するログを使用して「表 1.7-1 ログ出力の目的」に示す目的に沿ったログが出力されるよう設計すること。また，サブシステムの要件として，「表 1.7-1 ログ出力の目的」に示すログ出力の目的以外がある場合には，サブシステム毎に設計すること。 |

ログ出力の目的を下表に示す。

OS，プログラムプロダクト，アプリケーションが出力するログを使用して，下表の目的を達成できるように設計すること。また，サブシステムの要件として下表以外のログ出力の目的がある場合には，サブシステム毎に設計すること。

表 1.7-1 ログ出力の目的

| 項番 | 目的 | | 説明 |
|---|---|---|---|
| 1 | エラー，障害の対処 | エラー，障害検知 | ● 運用監視システムによるログ監視でエラー，障害の検知に使用する。 |
| 2 | | 原因解析 | ● システム管理者がエラー，障害時の原因解析情報として使用する。 |
| 3 | | リカバリ | ● システム管理者がエラー，障害時のリカバリ作業に使用する。 |
| 4 | キャパシティ管理 | 障害の予兆察知 | ● 障害の予兆察知のための情報（処理性能，リソース残量等）として使用する。 |
| 5 | | 設備条件整理 | ● 設備をスケールアウト又はスケールアップする際のサイジングの元データとして使用する。 |
| 6 | 監査，セキュリティ | | ● システムへのリクエストや処理内容を記録し，監査の証跡として使用する。<br>● セキュリティ違反の検知や処理内容把握，経路追跡のために使用する。 |
| 7 | SLA遵守確認 | 可用性管理<br>（稼動実績測定） | ● 稼動率，障害回復時間の非遵守回数等を計算するための元データとして使用する。 |
| 8 | | 性能管理<br>（応答時間測定） | ● 処理の応答時間遵守率を計算するための元データとして使用する。 |
| 9 | | ジョブ正常稼動非遵守件数測定 | ● ジョブが正常に動作しなかった件数の取得のための元データとして使用する。 |
| 10 | システム改善 | | ● 機能の利用状況を確認し，機能改善や廃止の元となる情報として使用する。 |
| 11 | デバッグ | | ● プログラムが正しく動作しているかを開発者が確認するために使用する。 |

### 1.7.2 ログ出力情報

| | |
|---|---|
| 目的: | ログに出力する情報を定義することで，エラーや障害の対処，キャパシティ管理，監査，セキュリティ，SLA遵守確認，システム改善，デバッグといった目的の達成を効率的に行えるようにする。また，運用監視システムとの連携ルールを定めることにより，運用監視システムと運用監視の対象となるサブシステムとの間の認識齟齬を防止する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | OS，プログラムプロダクト，アプリケーションがログ出力する情報は，「表 1.7-2 ログ出力情報」の「ログに出力するべき情報」を参考に設計すること。 |

ログに出力すべき情報を下表に示す。
これらのログ情報は，OS，プログラムプロダクト及びアプリケーションを組み合わせて出力する。

表 1.7-2 ログ出力情報

| 項番 | 目的 | | ログに出力すべき情報 |
|---|---|---|---|
| 1 | エラー，障害の対処 | ● エラー，障害検知<br>● 原因解析<br>● リカバリ | ● エラー情報<br>エラー，障害発生時の状況，発生箇所，等<br>● 処理トレース情報<br>プログラムプロダクト，アプリケーションにおける処理の履歴，等<br>● サーバ間通信処理情報<br>サーバ間（内部サブシステム間連携及び外部連携）の通信履歴，等<br>● データベース処理情報<br>データベースの処理履歴，トランザクションの成功／失敗，データの更新整合性を担保するために必要なデータ更新前後の情報[9]，等<br>● データベース更新ログ<br>DBMSが管理する更新ログ，等 |
| 2 | キャパシティ管理 | ● 障害の予兆察知<br>● 設備条件整理 | ● サーバ間通信処理情報<br>サーバ間（内部サブシステム間連携及び外部連携）の通信履歴，処理時間，等<br>● データベース処理情報<br>データベースの処理履歴，処理時間，等<br>● リソース情報等<br>ディスクやメモリ等の残量等の情報，等 |
| 3 | 監査，セキュリティ | | ● サーバ間通信処理情報<br>サーバ間（内部サブシステム間連携及び外部連携）の通信履歴，等<br>● サーバ機器のイベント情報<br>サーバの起動及び停止，OS／プログラムプロダクトへのログイン及びログアウト，特定リソースへのアクセス等，セキュリティポリシー上，取得保管が義務付けられるログ，等 |
| 4 | SLA遵守確認 | ● 可用性管理<br>（稼動実績測定）<br>● 性能管理<br>（応答時間測定） | ● サーバ機器のイベント情報<br>サーバの起動及び停止，等<br>● サーバ間通信処理情報<br>サーバ間（内部サブシステム間連携及び |

[9] データ不整合が発生した場合は，ログを利用してデータの整合性を取るが，データの整合性を担保するには，アプリケーションが出力するログだけでなく，SQLログやBPMS等，プログラムプロダクトのログも必要である。よって，プログラムプロダクトからもデータ整合性を担保するために必要なログを出力するよう設定する。

| 項番 | 目的 | | ログに出力すべき情報 |
|---|---|---|---|
| | | ● ジョブ正常稼動非遵守件数測定 | 外部連携)の通信履歴, 処理時間, 等<br>● 運用監視システムのジョブ管理エラー情報<br>ジョブ処理のエラー履歴, 等 |
| 5 | システム改善 | | ● 利用状況履歴<br>画面アプリケーションの機能利用履歴,<br>サービスの利用履歴, 等 |
| 6 | デバッグ | | ● デバッグ情報<br>開発者がプログラムの動作確認をするためにプログラムに組み込んで出力した情報, 等<br>● 「エラー, 障害の検知」,「原因解析」の目的で出力したログ |

### 1.7.3 ログレベル

| | |
|---|---|
| 目的: | ログレベルを統一的に定めることにより，ログ解析を効率化する。また，オペレーションベンダ等の習得コストを低減する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | 出力するログのログレベルは，「(2)A.ログレベルに関わる指針」に従って設計すること。 |
| 規約1: | 出力するログのログレベルは，「(2)B.ログレベルに関わる規約」のとおりとすること。 |

(1) ログレベルの定義
ログの重要度を示すログレベルの定義を下表に示す。

表 1.7-3 ログレベル

| ログレベル | 説明 | ログ出力文字列 | レベルの高さ | 通常運用時に出力する対象 |
|---|---|---|---|---|
| 異常 | システム管理者に通知し，エラーの調査や復旧のために操作が必要な例外・エラー（システムエラー，業務処理異常）が発生した場合に使用する。 | ERROR | 高 | ○ |
| 警告 | 利用者の再実行等で復旧可能な例外・エラー（タイムアウト等の業務的な警告等）が発生した場合に使用する。 | WARN | | ○ |
| 情報 | 本番環境での通常運用時に，システムの動作状況に関する情報（システムの開始・終了時，読み込んだ設定ファイルの情報等）を出力する際に使用するもの。 | INFO | | ○[10] |
| デバッグ | 開発環境や本番環境での動作確認時や異常が発生した際の解析時に，システムの動作状況に関する詳細な情報（プログラムのデバッグ情報としての，メソッドの入出力情報，処理の分岐の情報等）を出力する。 | DEBUG | 低 | × |

(2) ログレベルに関わるルール
A. ログレベルに関わる指針
ログレベルに関わる指針を以下に示す。

- 出力するログは，「(1)ログレベルの定義」に示すログレベルに分類して設計すること。
- 「情報」レベルのログは，ログ出力量が多くなりやすい傾向にあるため，ログ出力の目的を達成する必要最小限の出力量になるよう設計すること。
- 以下に示すログメッセージは，通常運用時には出力しない，「1.7.4 運用監視システムとの連携」にて示される監視対象のログファイルとは別のファイルへ出力する，等，ログ出力を抑止する制御を行うこと。
  - 処理の開始や終了，検索結果における該当データなし，等，アプリケーションの動作状況を単純に表しており，運用対処が無条件に不要であるログメッセージ。
  - ユーザが画面操作中に発生する入力チェックエラーを示すログメッセージ（ユーザ側に通知されて対処が行われ，別途運用対処が不要であるため）。
  - 様々な機能から呼び出される共通機能は呼び出し側にエラーコードや例外を返却し，呼び出し側が返却値に応じたログメッセージを出力することが望ましいため，共通機能側のログメッセージは出力抑止の対象とする。
  - 一つのメッセージを共通的に使い回すことは避け，運用対処が不要な事象に関する内容はログメッセージの出力抑止の対象とする。
  - 上記に当てはまらずログメッセージの出力対象としたものについても，運用開始後のログメッセージの出力状況（出力頻度，運用対処の緊急度と必要性，等）を定期的にチェックし，ログ出力の妥当性を確認すること。

[10] 一部のログはパフォーマンス調査等の目的で一定期間出力したり停止したりする場合を許容する。

B. ログレベルに関わる規約

ログレベルに関わる規約を以下に示す。

- デバッグ情報をエラー解析に利用したい等，出力するログのレベルを切り替えたい場合，設定ファイル等を変更することで，出力するログレベルを切り替えられるようにすること。

### 1.7.4 運用監視システムとの連携

| | |
|---|---|
| 目的: | 運用監視における重要度を統一的に定めることにより，重要度に応じた対処を効率的に行うことが可能となり，運用監視のコストを低減する。また，運用監視システムとの連携ルールを定めることにより，運用監視システムと運用監視の対象となるサブシステムとの間の認識齟齬を防止する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 指針1: | OS，プログラムプロダクト，アプリケーションが出力するログと運用監視システムとの連携は，「(2)運用監視システムとの連携ルール」に従って設計すること。 |

OS，プログラムプロダクト，アプリケーションが出力するログが運用監視の対象となる。運用監視システムとの連携ルール等を以下に示す。

(1) 運用監視システムとの連携方法と重要度

運用監視システムは，重要度に応じて予め設定された特定のキーワードがログに出力されたことを検知し，システム管理者へ異常等の情報を通知する。

運用監視における重要度は，『運用管理エージェント登録ガイドライン』に定義されている。下表に抜粋を示す。

表 1.7-4 運用監視における重要度

| 重要度 | 説明 | 例 |
|---|---|---|
| 重大 | 複数システム又は複数端末へ影響があり即座に復旧(リカバリ)を実施する必要があるエラー | ● ノードダウン<br>● クラスタ切替発生 |
| 大 | 単一システム内又は特定の端末のみで影響があるエラーで即日中に対処を実施する必要があるエラー | ● プロセスダウン<br>● プログラムプロダクトエラー<br>● アプリケーションエラー |
| 中 | 業務影響のないエラーで対処要否を判断する必要があるエラー | ● リソース閾値超過<br>● 管理者ログイン |
| 小 | 業務影響のないエラーで対処の必要がないエラー，又は，システム状態の通知 | ● 業務量多の警告<br>● 故障回復通知(リンクアップ)<br>● サービス起動通知 |
| 無 | エラーではないメッセージ | ● ジョブの起動・正常終了 |

(2) 運用監視システムとの連携ルール

運用監視システムとの連携ルールを以下に示す。

- ログレベル等のログ内容を使用することで，重要度に応じたキーワードを運用監視システムに設定できるようにログ出力内容を設計すること。
- 運用開始後における監視対象の変更は，運用監視システムのキーワード設定を変更することで，柔軟に対応し，極力，アプリケーション側のログ出力機能を変更しなくてもよいものとすること。

### 1.7.5 ログ出力項目とフォーマット

| | |
|---|---|
| 目的: | ログフォーマットを統一することにより，ログ解析を効率化する。また，オペレーションベンダ等の習得コストを低減する。また，運用監視システムとの連携ルールを定めることにより，運用監視システムと運用監視の対象となるサブシステムとの間の認識齟齬を防止する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 規約1: | ログ出力項目及びフォーマットは，「(1)ログ出力項目及びフォーマットのルール」のとおりとすること。 |
| 指針1: | 個別ヘッダ及びメッセージは，「(2)個別ヘッダ及びメッセージのルール」に従って設計すること。 |

(1) ログ出力項目及びフォーマットのルール

ログ出力項目及びフォーマットのルールを以下に示す。

- 運用監視システムとの連携を考慮し，1行で出力すること。運用監視システムが監視しない情報で，かつ可読性が向上する場合，改行を行うこと。
- 下表に示すログ出力項目とフォーマットに従い，ログ出力すること。

表 1.7-5 ログ出力項目とフォーマット

| 種別 | 項目 | 説明とフォーマット | 例 |
|---|---|---|---|
| ヘッダ | | ログの属性を表す項目を出力する。<br>フォーマット:<br>● ヘッダの1項目毎に角括弧(“[”及び“]”)で囲むこと。<br>● 項目間はタブを入れること。 | — |
| | ログ出力日時 | ログ出力日時(年月日時分秒ミリ秒)を示す。<br>フォーマット:<br>yyyy/MM/dd△HH:mm:ss.SSS | [2014/10/20△01:02:03.456] |
| | ログレベル | 「表 1.7-3 ログレベル」で定義したログレベルを示す。同表の「ログ出力文字列」に示す文字列を出力する。<br>フォーマット:<br>● 5文字で出力すること。<br>● 文字長が5文字未満の場合，語尾をスペースで埋めること。 | [INFO△] |
| | 個別ヘッダ(1つ又は複数) | エラーや統計等，対象のログを特定するための情報。ログ出力の目的毎に出力する項目が異なる。項目名と内容で構成される。定義済みの個別ヘッダの項目名は「表 1.7-6 定義済み個別ヘッダ」を参照のこと。<br>フォーマット:<br>「項目名」:「内容」 | [txid:1234567890] |
| メッセージ | | ログ出力の目的に応じて，適切な内容を出力する。 | 処理が開始されました。 |

※△は半角スペースを表す。

出力されるログのイメージを以下に示す。

[2014/10/20△01:02:03.456] [INFO△] [txid:1234567890] [proc:方式審査処理] 処理が開始されました。

※△は半角スペースを表す。また，項目間の区切りはタブである。

図 1.7-1 出力されるログのイメージ

予め定義済みの個別ヘッダを下表に示す。

表 1.7-6 定義済み個別ヘッダ

| 個別ヘッダ | 項目名 | 説明 | 例 |
|---|---|---|---|
| トランザクション識別子 | txid | 「エラー，障害の対処」，「監査，セキュリティ」等の目的でログ解析する際，複数のサーバのログを横断して，処理の履歴が追跡できるように出力した文字列のこと。同じ処理開始契機で動作したログに同一の識別子を出力する。<br><br>トランザクション識別子の発生方式と引き継ぎ方式を設計する必要がある。「エラー，障害の対処」，「監査，セキュリティ」を目的とするログで複数のログに横断的に出力される処理の場合は，必須で出力すること。 | [txid:1234567890] |
| 処理名／ID | proc | どこで発生した事象かを示す。<br>処理名称，処理モジュール名，画面名やそのID等 | [proc:方式審査処理] |
| 起動契機情報 | ユーザIDの場合：uid<br>ジョブIDの場合：jid | 起動契機になった処理の情報を示す。<br>画面の場合はユーザID，バッチの場合はジョブIDやディレード処理要求ID等 | [uid:user001] |
| 通信元／先情報 | 通信元の場合：src<br>通信先の場合：dst | 受信時の通信元や送信時の通信先の情報を示す。<br>URLやホスト名等 | [dst:http://xxxx/yyyy] |
| 接続先データベース情報 | db | 接続先のデータベースを示す。<br>接続先のデータベース名やID等 | [db:共有DB01] |

(2)　個別ヘッダ及びメッセージのルール

個別ヘッダ及びメッセージのルールを以下に示す。

- 個別ヘッダ及びメッセージの内容は，セキュリティの観点から，個人情報，非公開情報は出力しないように設計する。
- 定義済み個別ヘッダ以外が必要な場合，ログ出力の目的に応じて，サブシステム毎に設計すること。

### 1.7.6 ログ運用

| | |
|---|---|
| 目的: | ログ出力単位や保持方法などを統一的に定めることにより，ログファイルの取得及び参照を効率化する。また，オペレーションベンダ等の習得コストを低減する。 |
| スコープ: | 階層定型化サブシステム及びインタフェース定型化サブシステム |
| 規約1: | ログ出力媒体は，「(1)ログ出力媒体」に示す媒体とすること。 |
| 規約2: | ログ出力単位は，「(2)ログ出力単位」に示す単位とすること。 |
| 特例1: | 「(2)ログ出力単位」に示す特例を許容する。 |
| 指針1: | ログローテーションは，「(3)ログローテーション」に従い，設計すること。 |
| 指針2: | バックアップは，「(4)バックアップ」に従って設計すること。 |
| 指針3: | ログの参照用保持は，「(5)参照用保持」に従って設計すること。 |
| 指針4: | ログの参照用権限は，「(6)ログ参照権限」に従って設計すること。 |

(1) ログ出力媒体

ファイルに出力すること。

(2) ログ出力単位

ログの出力単位は，以下のとおりとする。

- ログ出力の目的ごとに，異なるファイルに出力すること。また，可読性や，後述する「(4)バックアップ」の頻度，「(5)参照用保持」の期間を考慮して，より細かい単位に出力先ファイルを区別してもよい。
- 複数のプロセスが1つのファイルに対して同時にログ出力しないこと。

複数プロセスから1ファイルへの出力について，特例を以下に示す。

- ジョブ処理等，アプリケーションの動作時間帯が重ならない等の理由で，ログ出力処理が競合しない場合，複数プロセスで1ファイルに出力することを許容する。
- 複数プロセスで同時にログ出力する場合でも，1ファイルに出力する要件があった場合は，以下の条件を満たす場合は，許容する。
  - 排他制御等を設け，ログの損失なく，ログ出力，ローテーション等ができる。
  - 性能要件を満たすことができる。

(3) ログローテーション

バックアップの周期やログ参照の利便性から，ログローテーションは以下のように行う。

- ログファイルの容量上限値を設け，容量上限値を超過したら，ローテーションを行うこと。
- ログファイルの容量上限値は，運用設計の結果を反映できるよう，設定値で変更できるようにしておくこと。
- 原則，ログのバックアップ周期に合わせて，適切な周期でローテーションを行うこと。ただし，他の契機（OS／プログラムプロダクトの起動・停止時等）でローテーションを行うことを許容する。
- ローテーションしても運用監視サーバによる監視が継続できるよう，最新ログのファイル名は固定とし，ローテーション後の古いファイルはファイル名をリネームすること。
- ローテーション後の古いファイルは，日付や連番等を付与し，最新ファイルとの対応付けと時系列が分かるファイル名にすること。

  例: 最新ファイル名: logfile.log
  ローテーション後のファイル名: logfile.log.20140102102030

(4)　バックアップ

『バックアップ設計指針』に従い，バックアップを行うものとする。

本書で定めるログ出力の目的と『バックアップ設計指針』におけるデータ種別の紐付けを，以下に示す。

表 1.7-7 ログ出力の目的とデータ種別

| 項番 | ログ出力の目的 | | 『バックアップ設計指針』におけるデータ種別 |
|---|---|---|---|
| 1 | エラー，障害の対処 | ● エラー，障害検知<br>● 原因解析<br>● リカバリ | (4) ログデータ (A) 更新ログ<br>(4) ログデータ (B) 各種ログデータ |
| 2 | キャパシティ管理 | ● 障害の予兆察知<br>● 設備条件整理 | (4) ログデータ (B) 各種ログデータ |
| 3 | 監査，セキュリティ | | (4) ログデータ (C) セキュリティ関連ログ |
| 4 | SLA遵守確認 | ● 可用性管理（稼動実績測定）<br>● 性能管理（応答時間測定）<br>● ジョブ正常稼動非遵守件数測定 | (4) ログデータ (B) 各種ログデータ |
| 5 | システム改善 | | (4) ログデータ (B) 各種ログデータ |
| 6 | デバッグ | | 該当無し。 |

(5)　参照用保持

ログの参照保持については以下のように設計する。

- 「エラー，障害の対処」を目的としたログについては，エラー解析のしやすさを考慮し，参照しやすい保存箇所，保存形式で保持する期間を設けること。
  例: 3週間分のログはログ出力フォルダに保持する　等
- ディスク容量を考慮し，参照保持期間を過ぎたログを削除する仕組みを設けること。

(6)　ログ参照権限

「監査，セキュリティ」を目的にしたログは，不当な消去，窃取，改ざん，業務上の必要なくアクセス等されないよう，ログには適切なアクセス制御を行うこと。