

# SMARTS™ Decision Manager

テクニカルホワイトペーパー v2.0



## 目次

<b>SMARTS™ decision manager</b>	1
<b>Decision Management</b>	2
Decision Management の原則	2
最適なアプリケーションのタイプ	2
組織の役割	3
<b>SMARTS™ ケーパビリティ</b>	4
SMARTS™ コンポーネント	5
SMARTS™ デプロイメント	6
SMARTS™ 内部アーキテクチャ	7
<b>業務のエンパワー</b>	9
デシジョンの編集と管理	10
デシジョンロジックの特別な管理手法: Dynamic QA	12
デシジョンロジックの特別な管理手法: ステートマシン	13
デシジョンに予測モデルを活用	14
デシジョンのテストと最適化	14
» シミュレーションエンジンの利用	16
» チャンピオン / チャレンジャー実験	16
» リアルタイムのデシジョン・メトリクス	17
<b>ライフサイクルを介したデシジョンの管理</b>	18
<b>効率的なデシジョンの適用と配備</b>	21
シンプルな呼出しモデル	21
ストレスのない適用と配備	22
最速なエンジンの活用	23
<b>高可用性、スケーラブルな適用と配備</b>	24
<b>エコ - システムへの適合</b>	26
<b>セキュリティ</b>	26
» 認証	26
» アクセス制御	27
» プログラム操作のセキュリティ	28
» ストレージのセキュリティ	28
» トレーサビリティ	29
<b>ステータスチェック&amp;ロギング</b>	30
<b>データアクセス</b>	30
<b>リモートサービス・アクセス</b>	31
<b>ローカリゼーション&amp;ブランディング</b>	32
<b>最後に</b>	32



# SMARTS™ Decision Manager

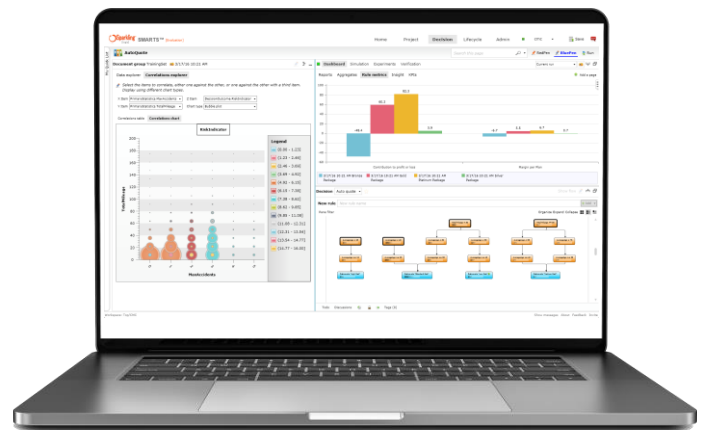
完成したDecision Management プラットフォーム

Sparkling Logic SMARTS™ は、下記機能の全てをサポートする「Decision Management」対応の統合開発プラットフォームです：

- デシジョンのモデリング（図化からプログラム自動生成）
- ビジネスルールのオーサリング（編集）
- 分析モデルをPMMLインタフェースを介してインポートし、運用できるようにする（Analytics/OPS）
- データから洞察ルールを導出し、ビジネスルールとして自動生し、適用・配備
- デシジョン分析機能でデシジョンの効率性を測定
- 目的や制約条件に応じたデシジョンの最適化
- 最適化されたデシジョンをクラウドおよび/またはオンプレミス環境にデプロイ（適用・配備）
- リアルタイムなデシジョン分析で、デプロイされたデシジョンによるビジネス成果をモニタリングとトラッキング可能
- 適用・配備されたデシジョンの継続的な改善・進化を動的にマネジメント

このホワイトペーパーは、統合、アーキテクチャ、デプロイ・オプションなど、SMARTS™ のより技術的な側面に焦点を当てています。対象の読者は、開発者、アーキテクトおよびその他の IT 専門家です。

ビジネスアナリスト向けの機能（デシジョンのオーサリング、テスト、シミュレーション）に関しては、別のホワイトペーパー「SMARTS Decision Manager , Analytics-Driven Automated Decisions 」をご参照ください。



## ● ● ● Decision Management

---

SMARTS™が、組織にどのように利益をもたらすかを知っていただくためには、まずは「Decision Management」とは何か？に関して、理解していただくことが大切かと思えます。

### Decision Managementの原則

Decision Managementとは、アプリケーションのロジックコード全体からビジネスロジックを分離してその部分を一元管理できるようにするアプリケーション構築のアプローチです：

- そのビジネスロジックは独立して、ITソフトウェアよりもビジネスに近い役割を担う人によって管理することができます。
- IT組織とは独立してテスト、最適化、デプロイ（配備）を行うことができるようになり、Decision Managementはビジネスのニーズに迅速かつ的確に対応することを可能とします。

Decision Managementは、最新のアプリケーションに期待される機能性や性能の水準をIT組織に任せながら、上記のように業務寄りのサービス機能/対応力の実現を可能にします。

### 最適なアプリケーションのタイプ

Decision Managementは、次のような課題を解決するアプリケーションにとっては、非常に有効な存在となります：

- ➔ 反復的なデシジョン及びアクションが、かなり多い場合
- ➔ 頻繁に調整が必要で、複雑な製品・サービス構成を対象とする場合
- ➔ 変化しやすいリスク要因がある場合

いろいろな事業領域で起きる上記のようなアプリケーション要件を「コンフィグレーション(構成)」という言葉で表現する場合があります。それは、クレジットベースの商品（クレジットカード、ローン、保険）、小売商品（スポーツ用品、専用車両）、さらには高価なデバイスやリソース（フォークリフトのバッテリー、配電）であっても同じです。また同様に、「リスク」という用語もさまざまな事業領域に適用され、金融リスク、健康リスク、デバイスの完全性リスクなどが含まれます。

Decision Managementは、上記のような特性を持つ多くのアプリケーション用途に対処しますので全ての業界で適用されています。

デシジョンが頻繁に、しかも大量に行われ、かなりのリスクを管理しなければならない場合にDecision Managementはその適用性を発揮します。Decision Managementの恩恵を受けるアプリケーションは、ミッションクリティカルなデシジョンである場合もあります。さらに、これらのデシジョンは静的なだけでなく、トレーサビリティや再現性などのさまざまな制約のもとで複雑なライフサイクルを経て、むしろ変化するものとなります。

Sparkling Logic SMARTS™は、長年にわたり、この種のアプリケーション開発に従事してきたビジネスルールとアナリティクスの専門家たちが「Decision Management」アプローチをサポートするためにチームとして集結し、ゼロから設計、開発した製品です。

## 組織の役割

Decision Managementは、組織内の関係者間の相互連携と協働を推進していきます：



**ビジネスアナリスト / データアナリスト / デシジョンマネージャー / デシジョンスペシャリスト / ルール・オーサー**は、通常、要件の把握、デシジョン・ロジックの実装、自動化されたデシジョンの品質管理を担当します。



**データサイエンティスト**は、ビジネスデータを分析し、ビジネスパフォーマンスに関するインサイトを抽出する役割を担っています。また、自動化されたデシジョンの品質を向上させるための予測モデルを開発することもあります。



**IT技術者/開発者/アーキテクト/エンジニア**は、自動化されたデシジョンが企業のインフラストラクチャ（IT基盤）に適合するかを確認し、また、デシジョン・サービスをアプリケーションに統合させることを行います。

このホワイトペーパーはIT技術者向けとしていますが、すべての役割をサポートするSMARTS™ Decision Managerの特徴全般について記載しています。。

## ● ● ● SMARTS™ ケーパビリティ

おおまかな理解としては、SMARTS™は多くのソース（図 1）を活用したデシジョンロジックの取り込みと実装ができるプラットフォームとなります：

- ビジネスエキスパート（専門家）の頭の中や、文書化される方針や手順
- ビジネスプロセスの一部として表現されるロジック
- 組織のデータソースに組み込まれたロジック（例えば、履歴処理データなど）
- 予測モデル/コードを実装したロジック
- これらのソースから発見・抽出されるロジックは、ルールやモデルとして SMARTS™ に実装され、SMARTS™ リポジトリに格納されます。

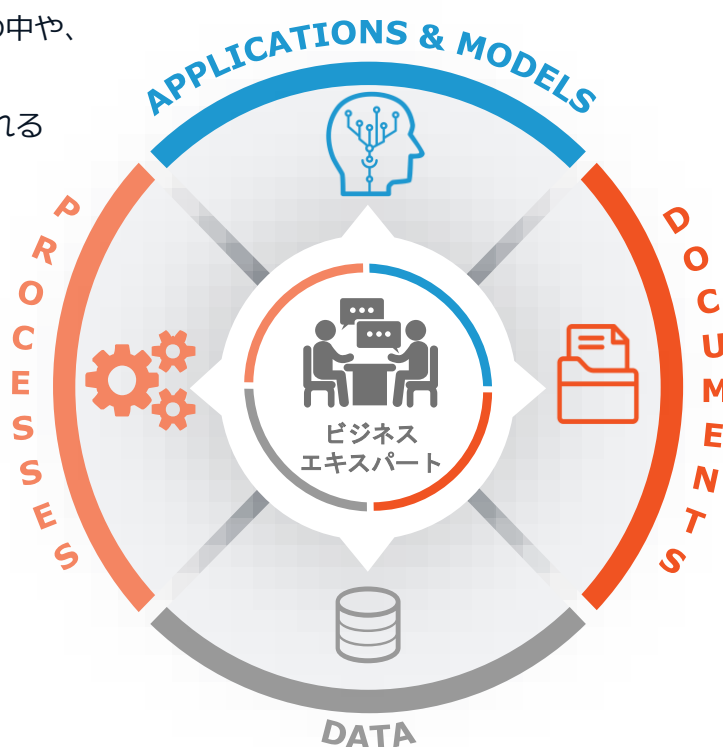


図 1 - ソース

デシジョンがリポジトリに一旦保存されると、以下のことが可能となります：

- デシジョンロジックを更新、テスト、最適化する作業が開始できます。
- デシジョンロジックを実行する自動化されたビジネスプロセス（タスク）を呼び出す本番システムに、デシジョンをデプロイします。
- IT部門が運用プロセスに介入することなく、ビジネスニーズに応じて更新を行いながら、そのライフサイクルを管理できるようになります。

# SMARTS™コンポーネント

SMARTS™ Decision Manager は、3 つの主要コンポーネントから構成されています (図 2) :

- 1 SMARTS™ Decision Repository** は、すべてのデシジョンアセット (ルールとモデル) を、内蔵のドキュメントデータベースに保存します。リポジトリは、アクセス制御、セキュリティ、監査証跡、変更、バージョン管理、リリース管理のトレーサビリティなどのガバナンスサービスを提供します。
- 2 SMARTS™ Decision Management環境**は、ビジネスルールや予測モデルを使用して自動処理可能なデシジョンをモデリング、テスト、シミュレーション、最適化が行えるビジュアルで直感的なWebベースのインタラクティブ・サービスを提供します。この環境は、デシジョンのオーサリングと管理を担当するビジネスアナリストやデータアナリスト向けに、HTML5/JavaScriptベースのZero-installユーザーインターフェイスとなっています。尚、同じこの環境で同タスクをプログラムで実行するためIT側にAPIを提供します。
- 3 SMARTS™ デプロイメント環境**は、リポジトリに保存されたデシジョンを実行可能にします。ビジネスプロセスと運用システムはデシジョンを行う必要がある時点で、評価・判定すべき対象データを渡して、デプロイメント (配備・展開) されたデシジョンを呼び出します。

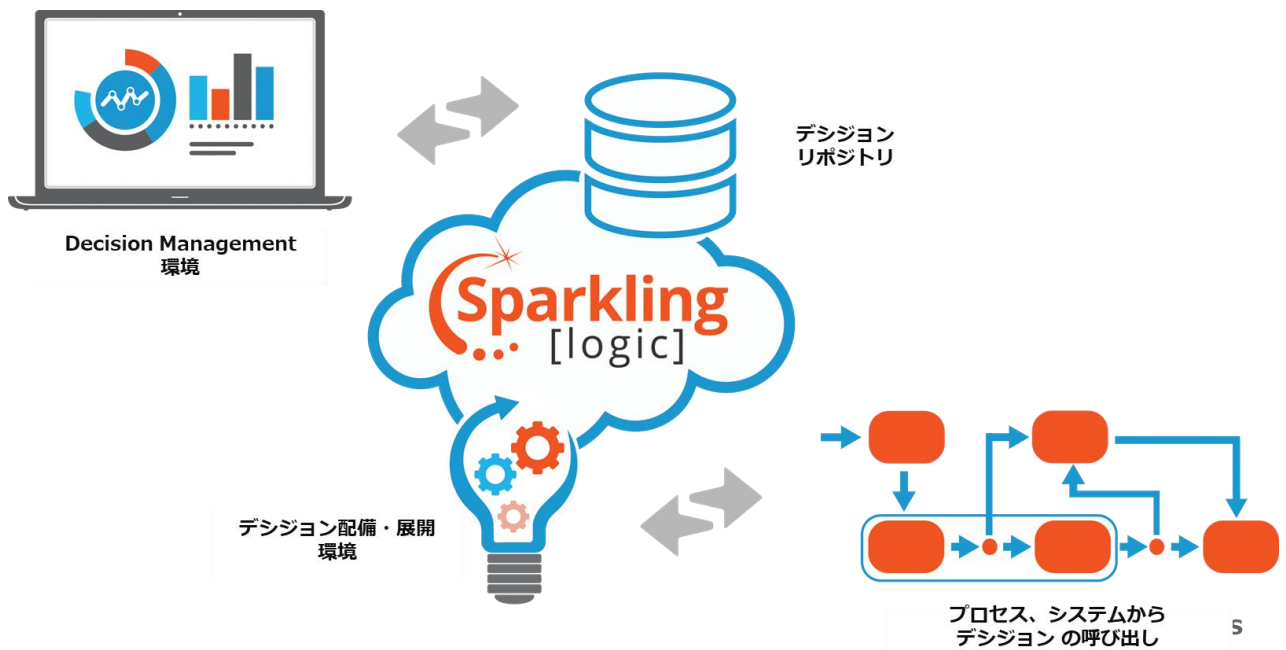


図 2 - SMARTS™ コンポーネント

## SMARTS™のデプロイメント

典型的な配備・展開は、認証された安全なJSON over HTTP(S) REST呼び出しで行うデシジョンサービスとなります。呼び出しは単純なリクエスト/レスポンスモデルに従います。リクエストは1つ以上のドキュメントに対するデシジョンの実行を要求し、レスポンスはその実行結果となります。

(SMARTS™ の用語として、ドキュメントの定義は、評価のためにデシジョンサービスに渡されるレコードまたはトランザクションのことで、つまり、ドキュメントは、デシジョンへの全ての入力とデシジョンからの出力を表します。SMARTS™ は一度にひとつのドキュメントを評価します。)

デプロイメントは、通常REST API を介してデシジョンサービスを呼び出すデシジョンのサービスとなります。SDKはJava、.NETフレームワーク、NodeJS、Python 3 等が利用可能となります。

ふたつめのデプロイメントオプションとしては、SMARTS™ デシジョンがアプリケーションと同じプロセスとスレッドで実行される組込み型デプロイメントです。このオプションは、外部からの呼び出しのオーバーヘッドに余裕がない、非常に大量や高性能を要求するアプリケーションに最適となります。この組込み型デプロイメントの場合、SMARTS™ デプロイメント環境は、Java または .NETコンポーネントとして提供され、呼び出しメカニズムはシンプルな Java または .NET API となります。

**SMARTS™ Decision Manager**は、クラウド対応とするために、ゼロから設計、開発されました。インストールの種類に応じて、いくつかの配信オプションを選択することができます。



**SaaSのお客様** (Sparkling Logic社がお客様に代わってクラウド環境を管理します) の場合、管理環境はWebブラウザのURIを通じて配信され、デプロイメント環境はアプリケーションがRESTを通じて呼び出すことができるデシジョンサービス呼び出しURIを通じて配信されます。



**クラウドサブスクリプションまたはオンプレミスのお客様** (お客様がクラウドまたはオンプレミスのインストールを管理します) の場合、管理環境とサービスデプロイ環境は、OVF 準拠の仮想アプライアンスまたはDockerイメージとして配信され、お客様のオーケストレーションツールを介してDockerコンテナとしてデプロイできます。

**すべてのお客様**には、アプリケーションに密結合できるJavaまたは.NET Frameworkまたは.NET Coreコンポーネントとしてデプロイ環境を提供することもできます。

Sparkling Logic SMARTS™デプロイメントは、レプリケートされたリポジトリを持つ複数のSparkling Logic SMARTS™インスタンスの前にロードバランサーを配置した最新のエンタープライズ高可用性とスケラビリティアプローチと同様に、最先端の最速な実行エンジン (ルールエンジン) を使用しています。



Sparkling Logic SMARTS™はエンタープライズ対応クラスの製品で、エンタープライズ環境との統合ポイントのいくつかを提供しています：

- ➔ リポジトリは、ユーザーの制御下でアルゴリズムとキーを使用して暗号化することができます。
- ➔ 認証は、法人向け認証プロバイダに委任することができます。
- ➔ 法人向けデータは、各種コネクタを介して 開発・テスト・最適化のためにアクセスすることができます。
- ➔ 法人向けサービスは、"リモート機能"でアクセスすることができます。
- ➔ Sparkling Logic SMARTS™のアクティビティストリームは、法人向けシステムに通知・プッシュされます。

## SMARTS™の内部アーキテクチャ

SMARTS™は、OVF 1.0に準拠した仮想アプライアンスとして提供され、仮想化環境（VMware、Oracle、Microsoft Hyper-V）で実行できるか、またはオーケストレーションツール（一般的にはKubernetes）を通じてデプロイできるDockerコンテナとして提供されます。UbuntuやAlpine Linuxのほか、Window Server CoreやNano Serverコンテナも提供されています。

ユーザーインターフェースによって操作され、実行時に活用されるすべてのアセットは、リポジトリ内で管理されます。リポジトリは、SMARTS™だけがアクセスできる完全組み込み型トランザクション文書データベースとして実装されています。

**このデータベースは、下記サービスを提供します：**

- ☑ **トランザクショナル・ドキュメント CRUD 操作** - すべてのアセットはリポジトリで管理され、すべての操作はトランザクションであり、有効な認証の下で実行されます。
- ☑ **バージョンング** - リポジトリ内のアセットへの変更は、すべてバージョンングされ、誰がいつ、なぜ変更したのかが追跡できます。
- ☑ **リリース管理** - デシジョンのライフサイクルの時点（通常はデプロイ時）で、リポジトリ内でのその後の操作に関わらず、変更がないことが保証された読み取り専用のリリースを作成することができます。
- ☑ **インデックス作成とクエリ** - 多くのインタラクションでは、アセットの関係から複雑なクエリを必要としますが、高度なインデックス作成とクエリ作成のメカニズムによってサポートされています。
- ☑ **インスタンス間のレプリケーション** - 多くのSMARTS™ インスタンスはプライマリ・レプリケーションまたはプライマリ・プライマリ構成でデプロイされ、高可用性とスケーラビリティを実現します。

- ☑ **バックアップ** - データベースは外部ディスク (VA) またはボリューム (Docker) としてマウントされ、エンタープライズツールを使ってバックアップすることができます。さらに、エクスポート/インポートのメカニズムも提供されています。
- ☑ **セキュリティ** - データベースとのすべてのやり取りは、SMARTS™の内部でhttps経由で行われ、アプリケーションからの直接のアクセスはありません。データベースはSQLを使用しないで純粋にAPIを通じて動作します。データベースは、顧客が管理する鍵で暗号化されている場合があります。



## DECISION LOGIC

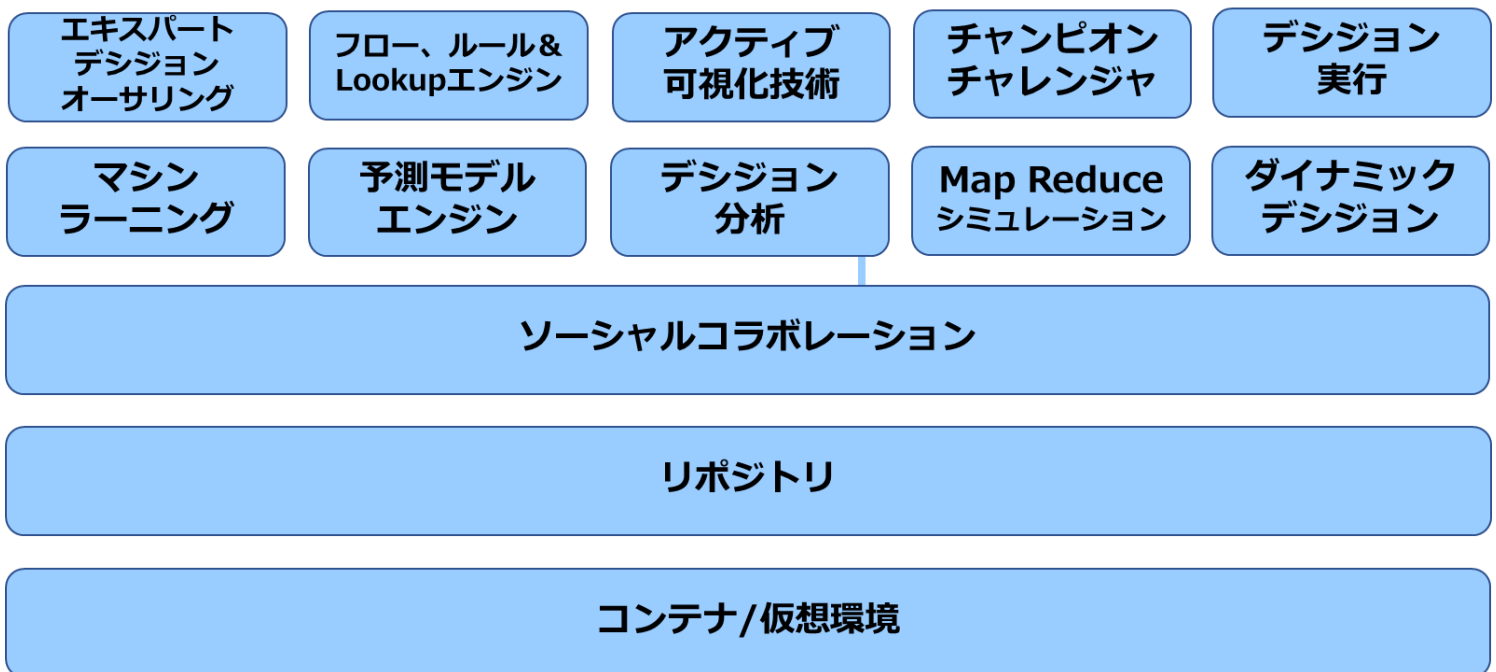


図 3 - SMARTS™ 内部アーキテクチャ

## ● ● ● 業務のエンパワー

Sparkling Logic SMARTS™は、民主化をめざしてビジネスアナリストを強力にサポートします：

- ➔ デシジョンロジックのオーサリングと管理
- ➔ デシジョンロジックのテストと最適化
- ➔ デシジョンロジックのライフサイクル管理

すべての活動およびそれぞれの関係者は、最新の、しかも同一Webブラウザベースのユーザーインターフェースが利用できます。同じユーザーインターフェースを使用することで、関係者間（ビジネスエキスパート、データサイエンティスト、デシジョンアナリスト、ITスペシャリスト）のミスコミュニケーションのリスクを軽減いたします。

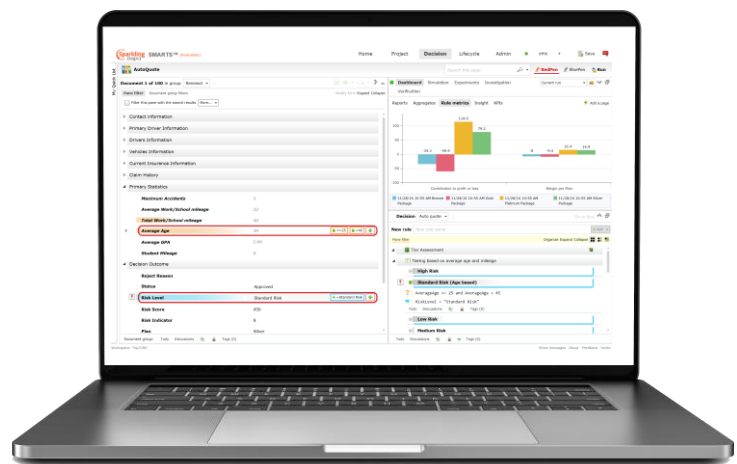
### このユーザーインターフェース：

☑ タブレットやスマートフォンを含むすべてのプラットフォーム上の最新ブラウザからアクセス可能で、ZeroインストールベースのWebアプリケーションを介して提供されます。

☑ 組み込みの認証プロバイダまたは法人プロバイダによる認証が必要となります。

☑ ロール（役割）ベースのアクセス制御メカニズムを使用しており、事前に定義されたいくつかの役割と、500以上の可能なコマンドへのアクセスが調整できる独自の追加機能を備えています。

☑ すべての操作がログとして記録され、主担当ユーザーには事前通知できるアクティビティ・ストリームを提供します。



## デシジョンの編集と管理

ビジネスアナリストは、さまざまなデシジョンのモデル化、実装、テスト、最適化を行うことができます。これらのデシジョンは、次のようなカテゴリに通常分類されます：

- ☑ **自動化されたデシジョン**：API 経由で渡されたデータに基づくデシジョンを、ストレートスルー（人手を介さずに一連の作業をシームレスに行う）でステートレスな方法でデシジョン・メイキングを行います。これらは最も一般的なデシジョンで、対話型およびバッチアプリケーションで使用され、最高レベルのパフォーマンスとスケーラビリティを必要とします。
- ☑ **ダイナミックな（質問に反射的に対応する）デシジョン**では、デシジョンロジックにより、最終ゴールとしてのデシジョンを対話的に、しかも自動的に導くようにユーザーにDynamic Questionnaire(ダイナミックなインタビュー対応)を行います。これは、最終的なデシジョンはエンドユーザーから取得した複雑なデータに依存します。つまり非常に複雑な製品構成の状況に見られるようなケースと似ています。反射的に対応しながら、次の質問へのデシジョンロジックは、以前の回答に基づいて取得したデータ要素を絞り込みながら、エンドユーザーにとって可能な限り最高のやりとりを行い、収集されるデータから得られる最高の品質を実現するために必要となります。
- ☑ **ステートフル・デシジョン**、他の処理が行われる間、複数のインタラクションでデシジョン・メイキングします。これらは、ストレートスルー型のデシジョンほど一般的ではなく、ステート間の遷移の一部として実行されるデシジョンロジックを持つステートマシンの組み合わせと見ることができます。これらは、デシジョンを多用するオーケストレーションやビジネスプロセスの実装によく使用されます。

Sparkling Logic SMARTS™は、これらのデシジョンをサポートしています。これらのタイプが圧倒的に一般的なので、このホワイトペーパーではSTP型処理のステートレスなデシジョンに焦点を当て、Dynamic Questionnaire（ダイナミックQA）とステートマシンについて紹介します。さらに、ビジネスアナリストは、SMARTS™ で管理すべきロジックに最適なデシジョン表現方法を活用することができます。これは、また作業中のデシジョンロジックの表現形式（メタファ）フォーマットをダイナミックに選択したり、切り替えたりすることもできます。

- ➔ デシジョンフロー
- ➔ ビジネスルール
- ➔ デシジョンテーブル、  
デシジョンツリー、デシジョングラフ
- ➔ ルックアップモデル
- ➔ 予測モデル

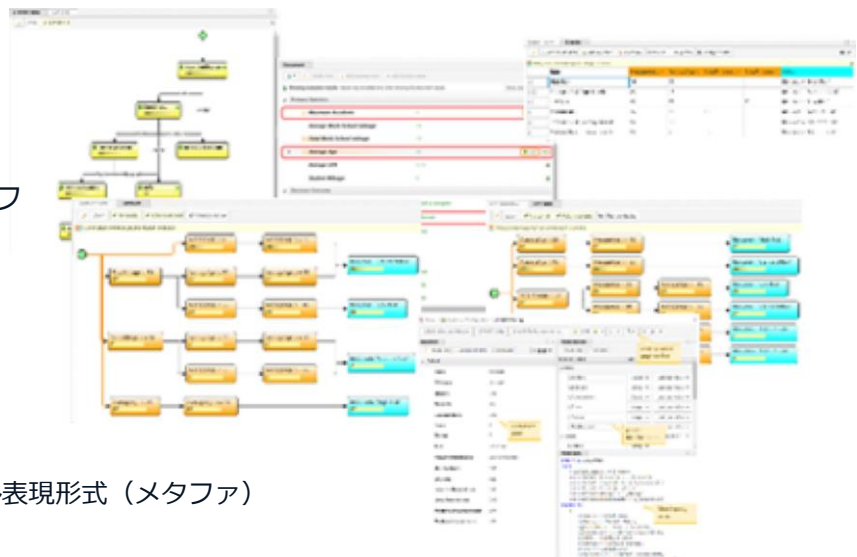


図 4- SMARTS™ グラフィカルルール表現形式（メタファ）

**Sparkling Logic SMARTS™**を使用すると、ビジネスアナリストは業務に最適な表現でデシジョンロジックを管理することができます。好みや手元の業務内容に応じて最適な表現（メタファ）を選択することができるからです。4種のグラフィカルなユーザーインターフェースにより、デシジョンロジックの作成と理解の両方を簡単に得ることができます。

このような表現の柔軟性は、実行の面で他の代償を伴うものではありません。SMARTS™ はマルチエンジンプラットフォームであり、デシジョンロジックの表現方法に応じて、最も効率的なエンジンを自動的に選択してくれます。これにより、ロジックを単一の表現（例えば、エクセルライク）に変換しなければならないというひとつのソリューションから解放されることとなります。

## **SMARTS™は以下の実行エンジンを提供：**

- デシジョンフロー実行エンジン
- ルールセットのデフォルトアクション、ルールセットおよびルールガード、優先度ルール、排他的および網羅的実行をサポートするシーケンシャルビジネスルールエンジン
- ビジネスルールエンジンとして推論をサポートし、Reteアルゴリズム発明者のDr.Charles Forgyが自身から提供した世界最速のReteアルゴリズム（Rete NT:New Technology）が実装されています。
- ルックアップモデルエンジンは、ルックアップデータを完全にインデックス化し、ルックアップルールを超高速で実行します。
- 12以上の予測モデルエンジンが、PMML標準に対応した予測モデルの実行をサポートします。

## **SMARTS™は、マルチステートのデシジョンロジックに対しても提供：**

- ☑ ステートマシン実行エンジン
- ☑ Questionnaire実行エンジン

このように多様な表現を用いることで、不要な構成要素がなく、余計なフロー制御がなく、ミスが少なく、何を実装しているのかが明確になり、役割間のコミュニケーションが容易になるなど、非常に効率的なデシジョンロジックの開発と維持が可能になります。同時に、複数の特殊な実行エンジンによる実装により、使用される表現形式に関わらず最適な方法で実行されることを保証します。

Sparkling Logicのお客様は、他のアプローチと比較して、SMARTS™のデシジョンはより少ないロジックでコード化され、実行速度が桁違いに速いことを確認されています。

## デシジョンロジックの特別な管理手法 (Dynamic Questionnaires : DQ)

前述のように、対話型の質問やインタビューシナリオに基づきエンドユーザーからデータを収集するデシジョンロジックの場合があります。例えば、生命保険の引受プロセスでは、エンドユーザーからのデータの収集には、数千にも及ぶ多くの潜在的な質問が必要になる場合もあります。ところが、実際にはそれほど多くの質問には回答することは不可能であり望ましいことでもありません。また、過去の回答に拠る質問もあるため、すべての質問が必要なのわけではありません。

質問ロジック (申込情報の収集) と自動化されたデシジョンロジック (生命保険商品の引受と構成) を1つのデシジョンの中で管理することが望ましい。これまで、ビジネスアナリストは対話型の質問ロジックとデシジョンロジックを分離して実装、テスト、最適化する必要がありました。質問ロジックとデシジョンロジックを分離すると、重複や矛盾が生じ、ITとビジネスの間に不協和が生じる可能性が出てくる場合が多くあります。

Sparkling Logic SMARTS™は、オプションのDynamic Questionnaire機能を提供しており、ビジネスアナリストは、自動化したデシジョンロジックと同じ環境で、質問全体を定義、実装、テスト、最適化することができます。また、同じロジック・アセット (既開発のソフト資産) を共有しながら、この種のアプリケーションの開発、テスト、デプロイ等を簡素化することもできるようになります。

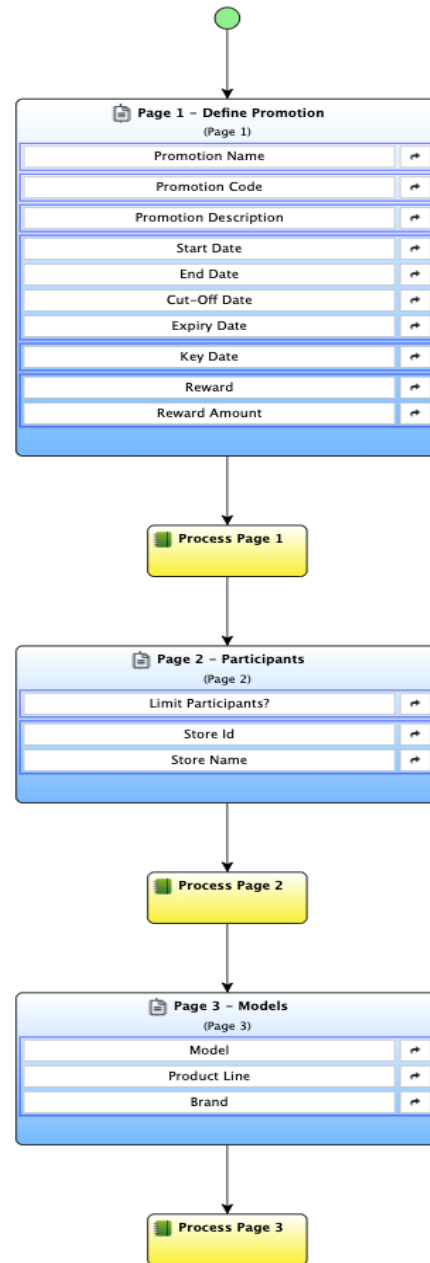


図 5 - SMARTS™のデシジョンフローと ダイナミック Questionnaire との動き

SMARTS™ で、ビジネスアナリストは、質問の設定、回答の処理、自動化されたデシジョンの実行が同じツールを使用して表現できます。ビルトインテストが行われるフローの中で、質問グループを表現するページからデシジョンロジックが定義できます（図5 参照）。また、各ページはエンドユーザーにとって、クライアントで実行される検証（意向確認等）とプレゼンテーション（商品紹介等）のロジックを保持できているので、最高の経験と最高のデータ品質を保証することになります。さらに、プレゼンテーションは標準的な CSS（Cascading Style Sheet）スタイルシートを使用してカスタマイズすることができます。

実行に関しては、質問はステートフルであり、SMARTS™ はサーバーと質問するクライアントの間で通信されるドキュメントの中にステートを保持します。このホワイトペーパーでは、ストレートスルーのステートレスなデシジョンのために同じデシジョン用の実行APIとメカニズムを使用していますが、ステートの永続性をどのように実現するかは、お客様が自由に決めることができます。

SMARTS™はAngularコンポーネントとして実装されたQuestionnaireレンダラーを提供し、どのようなWebアプリケーションにも組み込むことができます。さらに、完全なQuestionnaireプロトコルが仕様化（ドキュメント）されているので、必要に応じて独自のレンダラーを実装することもできます。

## デシジョンロジックの特別な管理手法（ステートマシン）：

特別なケースとして、複数のストレートスルー処理、ステートレスなデシジョンは、それらの呼び出しと追加サービスの呼び出しを組み合わせ合わせたオーケストレーション\*でつなぎ合わせる必要があります。例えば、クレジット・アプリケーションは以下のような場合があります：

- ➡ 申込の事前審査
- ➡ 事前審査に基づいて、取得すべき信用データの種類を決定する
- ➡ 信用データを収集する
- ➡ リスク評価を行う
- ➡ 謝絶、条件付き承認または承認/クレジット対象商品の構成管理を行う

\*：コンピュータシステム、アプリケーション、およびサービスにおける、設定、管理、調整の自動化を意味します。

このタイプのオーケストレーションは、Sparkling Logic SMARTSを含むテクノロジー等を活用して実装することができます。SMARTSは基本的にはステートマシン機能を提供しており、全体的なオーケストレーションフローを特別な状態で実行するアクティビティ（外部サービスの呼び出しを指定）として行うことで、SMARTS内で完全なオーケストレーション実装を実現できます。

業務担当はDynamic Questionnairesと同様に、他のサービスについてもSMARTS内で直接ステートマシンの実装、テスト、最適化で実現することが可能となります。

実行中のステートは、オーケストレーションクライアントとデシジョンサービスの間で通信されるデータに含まれています。ストレートスルーのステートレスデシジョンと同じAPIが使用されることになります。お客様は必要に応じたどの仕組みを利用しても、ステートの永続性を実現することができます。

## デシジョンに予測モデルを活用

SMARTS™ の特徴のひとつとして、データから得られたインサイト（洞察）をデシジョンロジックの実装に組み込むことができます。ビジネスアナリストは、SMARTS™に予測モデルを組み込むことで、信用リスクのデシジョン、マーケティングのネクストベストアクションやベストオファーのデシジョン、不正行為など、顧客の行動や不確実性に基づいたデシジョンを継続的に改善することができます。これは、SMARTS™が持つ2つの主要な機能によって実現できます：

### ● SMARTS™ BluePen™を活用して、説明可能で、エンジニアリング可能なモデルの創造

SMARTS™では、オプションのBluePenモジュールを活用し、大規模なデータセットを利用することで、説明可能なモデルを作成することができます。BluePenによって作成されたモデルは、ルールセット、テーブル、ツリーの表現に変換され、手書きしたルールのようにさらに改良・洗練することができます。専用の予測モデリングツールの代わりになるほどのものではありませんが、BluePenを使用することで、ビジネスユーザーは素早くモデルを作成し、デシジョンに反映、統合することができます。ビジネスアナリストはこの機能を利用して、データインサイトを利用してデシジョン結果を改善できるかどうかを探ることができます。

● 例えば、Python、SAS、SPSS、Rなどの予測モデリングプラットフォームで作成したモデルを、PMMLモデルとしてSMARTS™にインポートすることができます。PMMLは予測モードのための標準的なインターチェンジ・フォーマットです（詳細は<http://dmg.org> をご参照ください）。これらのモデルが SMARTS™ にインポートされると、どのデシジョンからも呼び出すことができます。これにより、ビジネスアナリストは、モデルがデシジョンの結果に与える影響を簡単にテストすることができます。SMARTS™ にインポートされたモデルは、呼び出したデシジョンと一緒にデプロイすることができます。SMARTS™ は、ニューラルネットワーク、多項式、一般回帰、線形/対数回帰、ツリー、サポートベクターマシン、ナイーブベイズ、クラスタリング、ルールセット、スコアカード、kNN、ランダムフォレスト、その他のマイニングモデルをサポートしています。

利用可能である外部サービスを呼び出して欲しいという状況がある場合があるかもしれません。このような外部サービスとは、予測モデルやデータプロビジョニングサービスかもしれません。SMARTS™ は リモート関数のサポートを提供しており、デシジョンの実行から JSON-RPC や REST サービスを呼び出すことができます。

## デシジョンのテストと最適化

デシジョンをオーサリングした後の重要な課題は、デシジョンをどのようにテストするかです。テストのサポートがなければ、ビジネスアナリストは、ロジックが正しいかどうかを評価するために、デシジョンロジックが適用・配備され、結果が取得されるまで待たなければなりません。これは、導入作業プロセス全体に大きな軋轢を与え、複雑なテストシナリオに対処したり、追加テクノロジーを投入することをIT部門に強いてしまうため、大きな問題となります。



テストを行い、その品質を定量的に把握することは、デシジョンの積分管理として不可欠な部分です。SMARTS™ では、デシジョンの品質を評価するためのデシジョンメトリクスが定義できます。メトリクスが作成されると、SMARTS™ は完全なシミュレーション環境を提供できることになり、同じ IDE (IT統合開発環境)で直接メトリクスによる計測を可能とします。デシジョンロジックのオーサリングや管理に使用すると同じツールや表現を使用してメトリクスを作成できるため、ビジネスアナリストの学習する手間が最小限に抑えられることとなります。

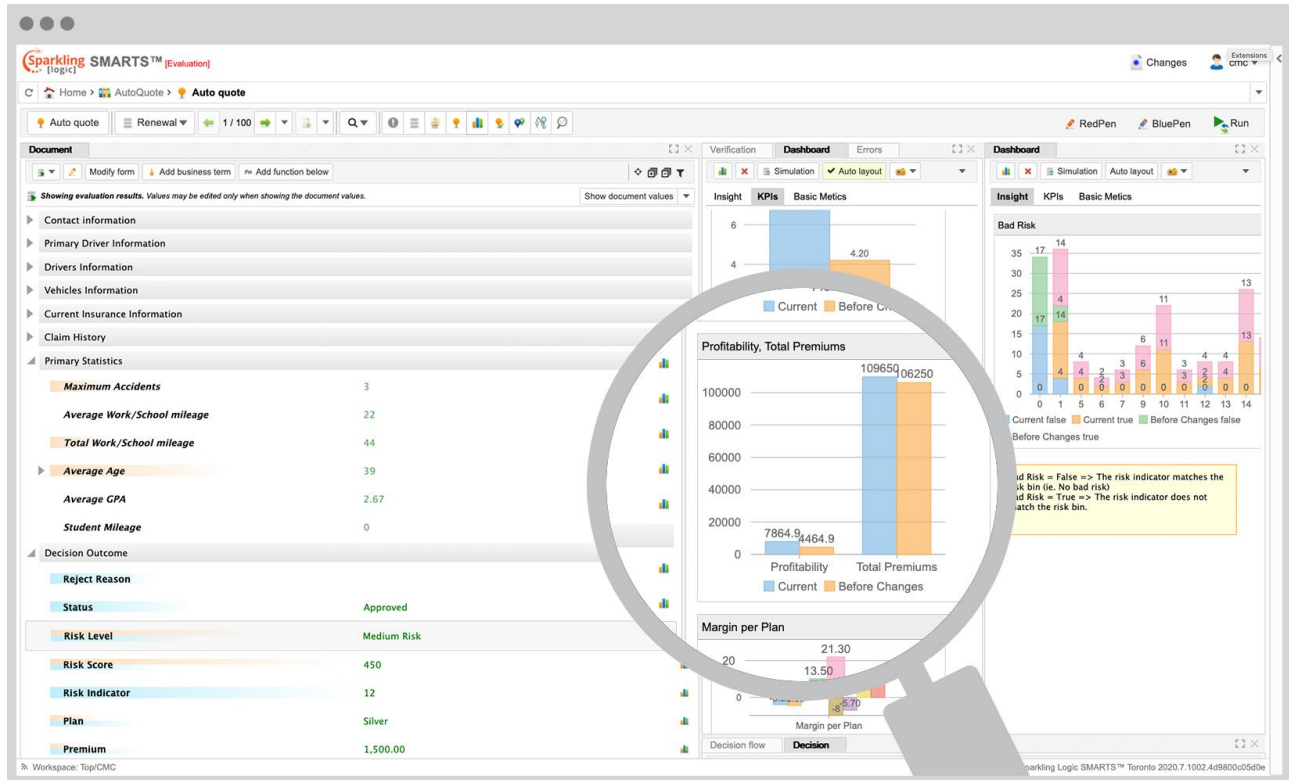


図 6 – SMARTS™ グラフィカルメトリクス対応のダッシュボード機能

あるビジネスアナリストは、デシジョンの質を別の側面から測定するために、異なるメトリクスを定義することがあります。多くの場合、理想的な組み合わせとしては、複数のメトリクス間の妥協またはトレードオフとなります。これらのメトリクスを利用できることは、デシジョンの品質をビジネスと IT の関係者間で連携を取るための貴重な手段となります。

これらのメトリクスを計算するには、データを選択したシミュレーションでデシジョンを実行します。データへのアクセスは、SMARTS™ の特徴の中核をなすものです。ほとんどの IT 組織は、データ管理のために一貫した標準とその管理の仕組みを維持したいと考えています。

IT 組織は SMARTS™ を使用し、以下を行うことでその目的の実現を可能にします：

- ビジネスアナリストが SMARTS™ 内での管理やシミュレーションに使用できるデータを SMARTS™ にアップロードして、データセットを設定することができます。
- SMARTS™ リポジトリ内に保存することなく、ビジネスアナリストがアクセスできるデータを、シミュレーションのために SMARTS™ を介してストリーム配信されるデータセットとして設定することもできます。

SMARTS™は、さまざまな表現でエンコードされたデータセットを認識できます。JSON、XML、CSV、PSV、およびAvroです。これらのデータセットは、様々なメカニズム：FTP(S)、SFTP、HTTP(S)、S3、RESTを介して、SMARTS™に提供されます。SMARTS™はシミュレーション用のデータをストリーミングすることもできるため、シミュレーションに使用するレコード数に制限はありません。つまり、シミュレーションに使用されるデータセットのサイズは問いません。

## シミュレーション・エンジンの利用

SMARTS™は、シミュレーションを迅速かつ効率的に実行できるように、ビジネスアナリストやIT担当者によるmappersやreducersの設定を行うことなく、すべてのCPUコアを利用して効率的に活用できるMap-reduce\*シミュレーションエンジンを利用しています（図7を参照）。

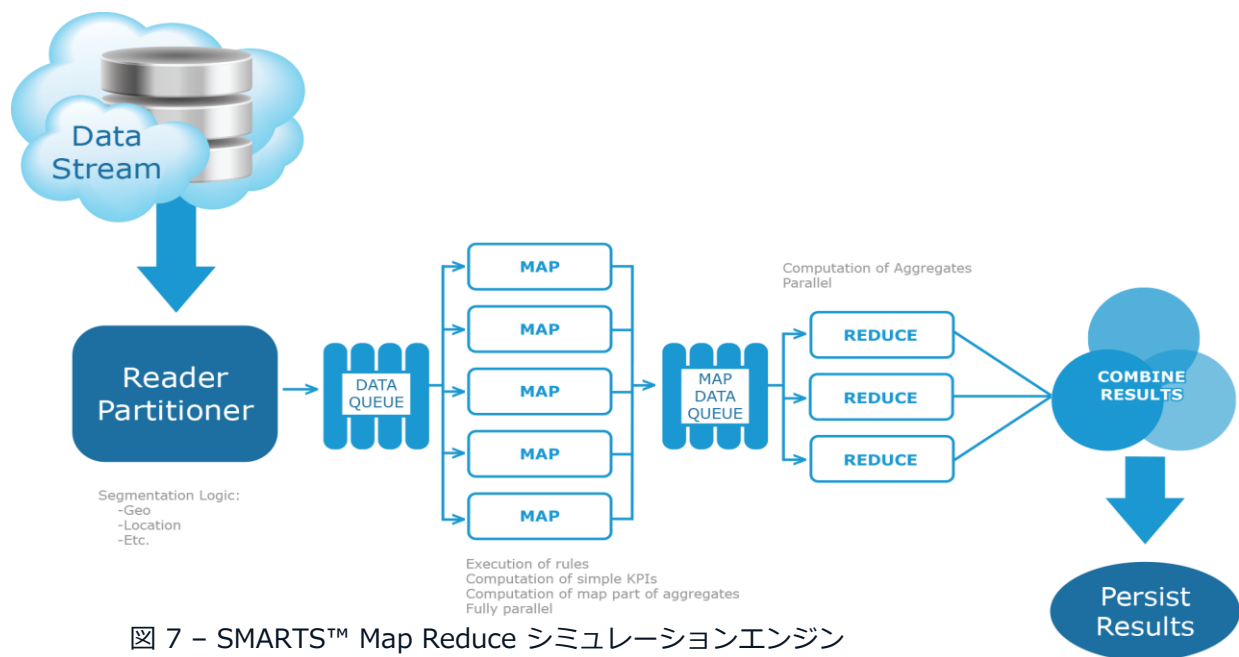


図 7 – SMARTS™ Map Reduce シミュレーションエンジン

\*:MapReduceは、ペタバイト規模のデータを小さなチャンクに分割し、Hadoopの汎用サーバーで同時に処理することで、並行処理を容易にします。最後に、複数のサーバーからのすべてのデータを集約して、統合された出力をアプリケーションに返します。

シミュレーションは、ユーザーインターフェースまたはAPIを介して起動することができます。シミュレーションの結果は、参照できるようにSMARTS™リポジトリに保存されます。

例えば、SMARTS™APIを使用して、開発中のデシジョンのシミュレーションを毎晩、実際のトランザクション（先月の実行処理）と比較してトリガーするcronジョブ（デーモンプロセス）を作成することができます。これらのシミュレーションは、開発中でデプロイする前のデシジョンの実パフォーマンスを具体的にモニターすることに使用できます。

## チャンピオン/チャレンジャー実験

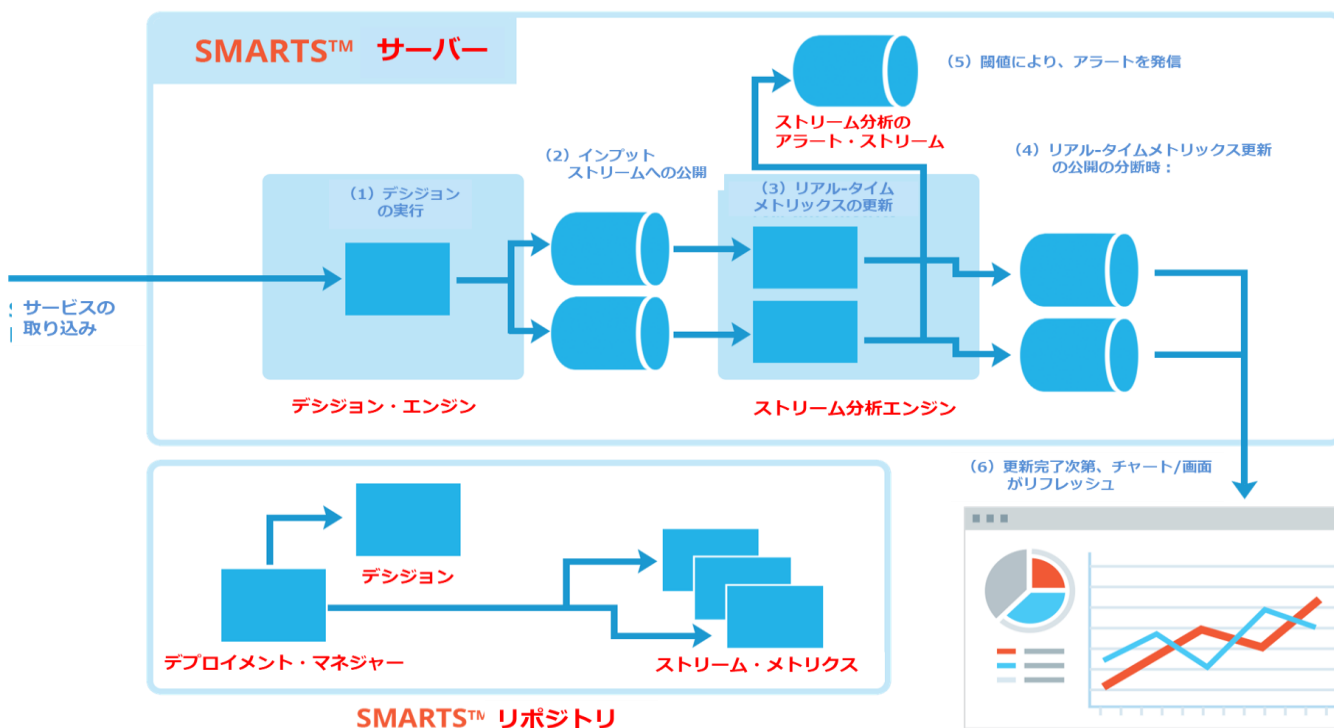
また、ビジネスアナリストがデシジョンロジックの改善案を思いついても、それが導入されるまでテストできないということもあります。例えば、顧客プロファイルに基づいた割引のマーケティングオファーを想像してみてください。何人の顧客が割引の対象となるかはわかるが、何人の顧客が割引に応じるかはわからない。

SMARTS™ を使用して、デシジョンのごく一部を改良したデシジョンロジック(チャレンジャー)に、残りの部分を既存のデシジョンロジック(チャンピオン)を適用した実験を行うことができます。このアプローチは、A/B テスト、Champion / Challenger (チャンピオン/チャレンジャー)、または Experimental Design (実験計画) と呼ばれています。

チャンピオン/チャレンジャー実験では、デプロイされたデシジョンが本番データに適用されたか(チャレンジャーか?チャンピオンか?)を追跡し、時間の経過とともに対応された品質を測定して、現在のチャンピオンに代わってチャレンジャーのいずれかを推進するかどうかの判断をするのに重要となります。

## リアルタイムのデシジョン・メトリックス

SMARTS™ を使用すると、ストリーミング分析を使用したリアルタイムのデシジョン・メトリックスを実装することができます。SMARTS™ はリアルタイムでメトリックスを更新するので、既存のデシジョン(Champion)に対する改善されたデシジョン(Challenger)の相対的な品質を知ることができます。簡単な設定でリアルタイムのメトリックスを指定できます。外部ツールは必要ありませんし、他のインフラとの連携も必要ありません。



Copyright ABC協会 (PMAI会) All Right Reserved

図 8 - SMARTS™ ストリーム分析エンジン

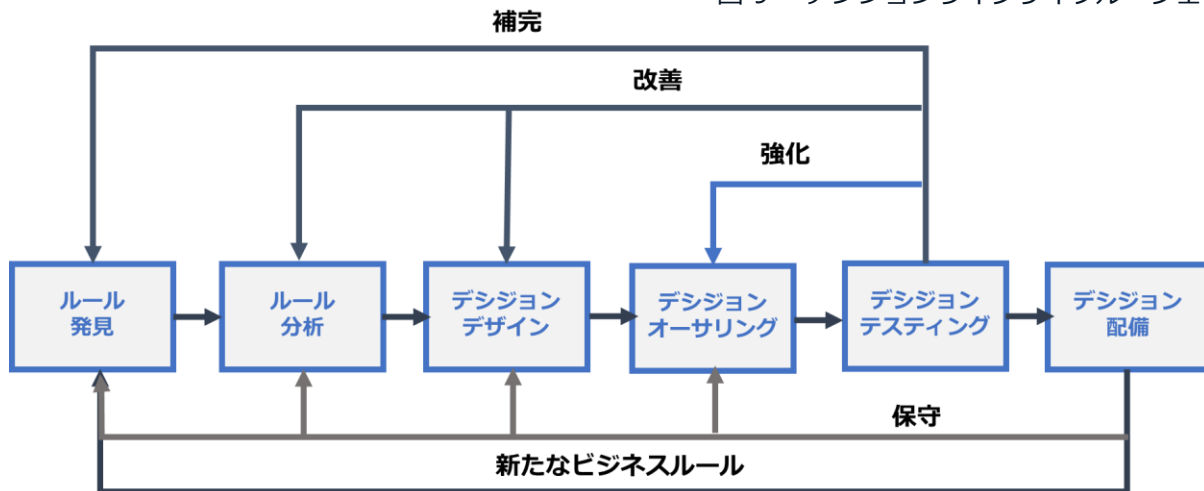
SMARTS™ リアルタイムのデシジョン・メトリックス機能はChampion Challengerの他にも適用されます。リアルタイムのデシジョン・メトリックスをどのようなデシジョンにも適用することで、実行中のデシジョンの品質を監視することができます。この機能を使用すると、予想からの異常な逸脱を検出することができますし、さらに逸脱が発生した時のトリガーをかけるアラート(警告)を定義することもできます。これらのアラートは、企業の通知システムに連携することもできます。

## ● ● ● ライフサイクルを介したデシジョンの管理

ライフサイクルの中で、デシジョンは下記のような複数のステージを経て進化していきます：

- ☑ 動機、発想、発見
- ☑ 分析と設計
- ☑ オーサリング（登録・編集）と実装
- ☑ テスト
- ☑ 改善・最適化
- ☑ デプロイ（配備）
- ☑ リタイアメント（削除）

図 9 - デシジョンライフサイクル・フェーズ



これらのステージはリニアではなく、最初の配備後、実装、テスト、最適化のサイクルを繰り返しながら、デシジョンが何度も見直されます(図 9 参照)。

SMARTS™では、お客様の要件を正確に満たすために、デシジョンのライフサイクル管理プロセスを構成しています。 SMARTS™ は、以下の機能を持ってライフサイクル管理をサポートします：

▶ **リリース** -ある時点の読み取り専用のデシジョン（スナップショット）で、デシジョンの参照セットアップとして機能し、テストやデプロイ時に変更されないことが保証されています。

▶ **ワークスペースへのリリースの公開** - SMARTS™では、あるワークスペースから別のワークスペースへリリースを公開することができます。 典型的なワークスペースは、テスト、ユーザー受入テスト、パフォーマンステスト、統合テスト、デプロイ/運用向けに定義されています。 これにより、(特定の役割にアサイン)ユーザと各環境に固有のプロセスを分離することができます。

▶レプリケートされたSMARTS™ インスタンスに役割を割り当てることで、ライフサイクルステージ毎にハードウェアリソースを用途毎に分離することができます。

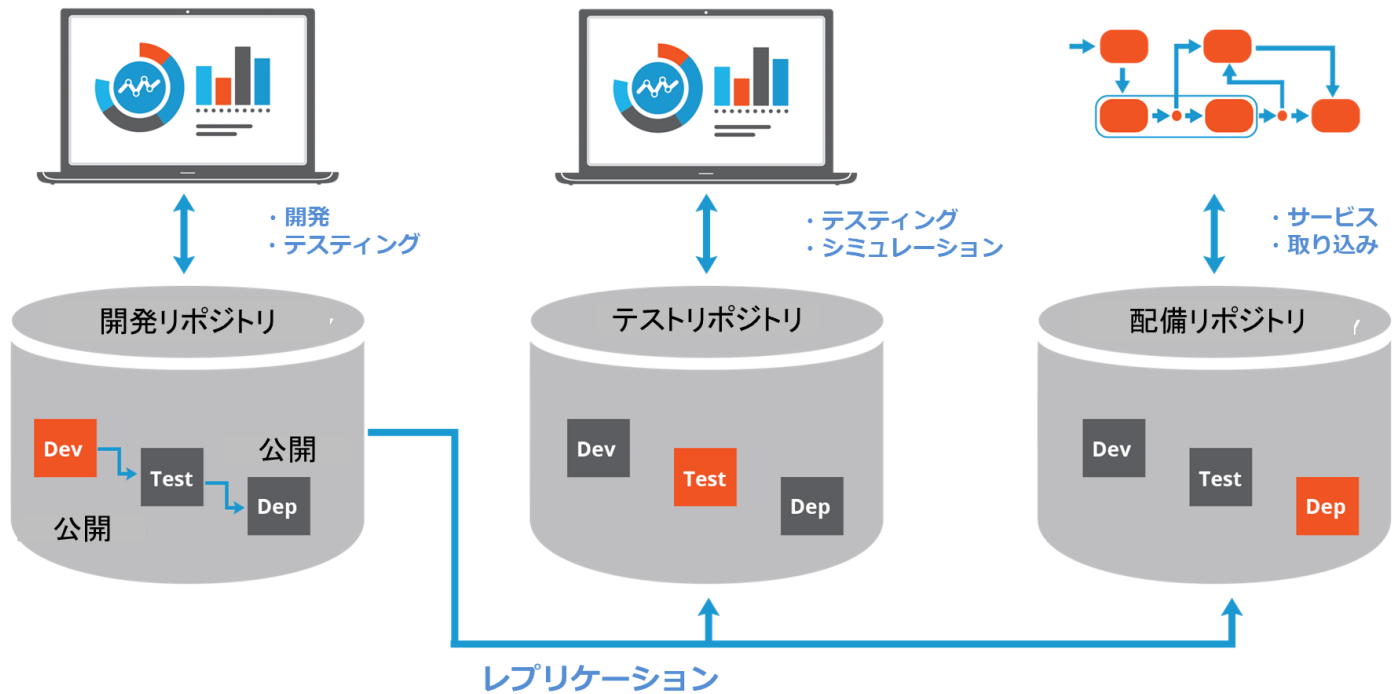


図 10 - リリースのレプリケーション

ワークスペース間のパブリッシュ操作とリポジトリのレプリケーションを組み合わせることで、図10に示すような完全なマルチマシンのライフサイクル管理フローを確立することができます：

- ➡ レプリケーションは、テストとデプロイメントのインスタンスのリポジトリを開発中のマスターと同期させておくために使用されます。
- ➡ ワークスペースは、開発、テスト、およびデプロイメントのアクティビティを分離するために使用されます。
- ➡ パブリケーションは、開発からテストおよびデプロイメントのワークスペースにリリースをパブリッシュするために使用されます。

技術的な観点から言えば、このようなイメージを実現するために必要なのは、必要なレプリケーション・トポロジーに応じてインスタンス（アプライアンスやコンテナ）を適切に構成することだけです。これは、製品のユーザーインターフェースを介してグラフィカルに、またはインスタンス化する時に付与される設定仕様内容を介して行うことができる簡単な操作となります。

SMARTS™ は、すべてのライフサイクル管理操作をプログラムで実行できる API 機能を提供しています。例えば、この API 機能を使って、デシジョン資産のライフサイクルを、他のソフトウェア資産であるライフサイクルに結びつけられることができます。API は REST API であり、以下のような「デプロイメントフロー」を実装することで可能となります：

- ☑ 開発ワークスペースでリリースを作成する
- ☑ リリースをテストワークスペースに公開する
- ☑ リリースに対するテストワークスペースでのシミュレーションの実行
- ☑ シミュレーションの結果を確認し、期待値からの乖離があればフラグを立てる

## ● ● ● 効率的なデシジョンの適用・配備

---

デプロイされたデシジョンは、2つの重要な技術的側面を考慮しなければなりません：

- ☑ 実装が簡単で、アプリケーション・アーキテクチャに問題を与えないこと。
- ☑ 高い可用性とスケーラブルなシナリオをサポートする必要があること。

SMARTS™自体は、アプリケーション・アーキテクチャに問題を与えません。お客様のニーズや環境に応じて、同じデシジョンロジックを複数のモードでデプロイすることができます。さらに、SMARTS™は、高可用性とスケーラブルなシナリオを強力にサポートします。

### シンプルな呼び出しモデル

SMARTS™は、APIの複雑さを最小限に抑えるように設計されたシンプルなデシジョンを実行するインターフェイスを提供します。デプロイメントモードやプラットフォームに関係なくSMARTS™でデシジョンを実行するには、以下のことが必要になります：

- ➔ 実行するデシジョンを特定し、それを実行するためのアクセストークンを取得します。このアクセストークンは、期間が限られており(ユーザー定義)、リフレッシュトークン(ユーザー指定のオプション)を伴うことがあり、任意の時点で取り消される可能性があります。
- ➔ 同期実行APIを呼び出す；
  - ➔ デシジョンを実行するために1つ以上のドキュメントを渡すこと。ドキュメントは、ルールを指定するために使用される構造と互換性のある構造である必要があります。
  - ➔ 渡されたドキュメント毎に、結果ドキュメントとして1つ返します。各ドキュメントには、渡されたすべての内容に加えて、デシジョンロジックが変更または追加された内容が含まれます。

すべてのデプロイモードとプラットフォームは、このモデルに適合するAPIを使用します。このモデルはシンプルなので、デシジョンを定義するのが簡単です。デシジョンとは、処理の結果として提供されるドキュメントを完璧にするデシジョンロジックのことです。

SMARTS は呼び出し API にいくつかの追加機能を提供しています：

- ・ 相関 ID を指定して、後で入力と出力を一致させることができますようにします。
- ・ さまざまなレベルで、実行のトレース(フロー、ルール、モデルなど)を指定します。
- ・ ペイロードが非常に大きくなる可能性があるので、ペイロードを圧縮することを指定します。

## ストレスのない適用と配備

SMARTSは、複数のデシジョン・タイプのデプロイメントをサポートしています。リポジトリ内の同じデシジョンを複数のプラットフォームに、複数のモードでデプロイすることができます：

オプション	デプロイメントモード	特性	いつの時点？	プラットフォーム
<b>デシジョンサービス</b>	<ul style="list-style-type: none"> <li>・ レポジトリベース</li> <li>・ デシジョンファイルベース</li> </ul>	<ul style="list-style-type: none"> <li>・ SMARTSインスタンスを通じて疎結合されたセキュアな REST APIのデプロイ</li> </ul>	<ul style="list-style-type: none"> <li>・ 対話型アプリケーションの一部としてのデシジョンの提供時点</li> <li>・ サービス指向の環境の一部としてのデシジョンの提供時点</li> </ul>	<ul style="list-style-type: none"> <li>・ リポジトリベースのサーバー:VAまたは Dockerコンテナ (Linux,Windows)。</li> <li>・ デシジョンファイルベース: Dockerコンテナ (Linux,Windows, Java, .NETコア, .NET フレームワーク, NodeJS, Python 3の REST SDKを介した呼び出し。</li> </ul>
<b>デシジョンコンポーネント</b>	<ul style="list-style-type: none"> <li>・ デシジョンファイルベース</li> <li>・ コード生成ベース</li> </ul>	<ul style="list-style-type: none"> <li>・ 密結合された.NET, .NET Coreまたは Java API</li> <li>・ プロジェクトの生成,デシジョンアーカイブまたはコードを介して展開</li> </ul>	<ul style="list-style-type: none"> <li>・ バッチ処理の一環としてのデシジョン</li> <li>・ 非常に高いスループットが要求</li> <li>・ デシジョンは、密結合されたネイティブコンポーネントの一部</li> </ul>	<ul style="list-style-type: none"> <li>・ デシジョンファイルベース: .NETコア.NET フレームワーク。</li> <li>・ コード生成ベース: Java</li> </ul>
<b>デシジョンプロバイダーと実行ノード</b>	<ul style="list-style-type: none"> <li>・ デシジョンファイルベース</li> </ul>	<ul style="list-style-type: none"> <li>・ 疎結合されたPub/Subアーキテクチャ、</li> <li>・ デシジョンの提供、実行、エラーの処理ノードを使用</li> <li>・ Rabbit MQ</li> </ul>	<ul style="list-style-type: none"> <li>・ メッセージ重視のデシジョン</li> </ul>	<ul style="list-style-type: none"> <li>・ デシジョンファイルベース: RabbitMQ、クラウドフレームワーク pub-sub</li> </ul>



どのデプロイメント・オプションを選択するかは、アプリケーション全体のアーキテクチャ選択と同様に、アプリケーションのニーズの問題となります。

一般的には、

- ☑ デシジョンが頻繁に変更される対話型アプリケーションのデシジョン部分には、リポジトリベースのデシジョンサービスを使用します。
- ☑ オーケストレーションされたデプロイメントに重点を置いた大規模なマイクロサービスアーキテクチャのデシジョンの部分には、リポジトリまたはデシジョンファイルベースのデシジョンサービスを使用します。
- ☑ 大規模なバッチジョブ（数百万レコード以上）のデシジョンには、デシジョンコンポーネントを使用します。

SMARTS™ は、複数のデプロイメント・オプションをサポートしているため、お客様のアプリケーション環境に簡単に適合させることができます。

## 最速なエンジンの活用

先ほど強調しましたように、SMARTS™ は複数のエンジンを利用してデシジョン・メイキングを行い、それぞれが特定のタスクに適応しています。

シーケンシャルルールエンジンは、一般的なビジネスルールの実行に非常に効果的です。そのロジックはすべて、.NET Core、.NET Framework、またはJava上のバイトコードにコンパイルされ、インタープリタは関与しません。これにより、このエンジンは高速に実行することができます。このエンジンは、取引詐欺の業務適用において、4コアのIntel i7ラップトップ上で1秒間に30万件以上のトランザクションを処理するというベンチマーク結果がでています。この場合、このエンジンで、シナリオ内のビジネスルールの大部分を処理しています。

Rete-NT と呼ばれるエンジンは Rete 推論エンジンの最速バージョンです。すべてのルール表現は、.NETコア、.NETフレームワーク、またはJava上のバイトコードにコンパイルされ、アグレッシブなインデックス化を備えたRete実装により、推論アルゴリズムの実行が確実に行われます。従来の推論性能テスト（例えばWaltzDBファミリ）において、他のRete実装よりも10~100倍高速であることがベンチマークで確認されています。このエンジンは、推論が必要な場合や、使用するフィールドの数に対してルールの数が著しく多い場合には、最適な選択となります。

ルックアップモデル・エンジンは、ルックアップに関係するすべての式をバイトコードにコンパイルし、ルックアップデータを完全にインデックス化するので、検索が非常に高速になります。例えば、一連の基準に対応する薬剤の選択を実装した64Kのルールセットは、インデックス化されたスペースを1回の検索で終了します。

## 高可用性、スケーラブルな適用と配備

SMARTS™ デシジョンサービスは、エコシステム内での高可用性とスケーラビリティの両方をサポートします。サポートのレベルを説明するために、この章では、リポジトリベースとデシジョンファイルベースのデシジョンサービスについて説明します。

以下の例（図11）では、典型的なリポジトリベースの SMARTS™ デシジョンサービスのデプロイについて説明します：

- ➡ コンテナを介して実装された合計 9 個の SMARTS™ インスタンスが 3 つのノードにあります。
- ➡ この 9 つのインスタンスのうち、1 つはリポジトリマスタであり、開発とテスト環境の役割を果たしています。
- ➡ 他の 8 つのインスタンスはデシジョンの実行に専念しており、負荷を分散するインフラストラクチャの一部であるロードバランサを介してアプリケーションからアクセスされます。ロードバランサは、デシジョン実行インスタンスの健全性を保つように構成されています。
- ➡ これらの実行インスタンスのうち、各ノードに1つずつある3つの実行インスタンスは、リポジトリの複製コピーを持っています。すべてのインスタンスがダウンしてプライマリリポジトリのインスタンスが戻ってこなくても、これら3つのレプリケートされたリポジトリのおかげでデシジョンサービスは利用可能となります。
- ➡ 実行に専念する残りのインスタンスは、独自のリポジトリを持たず、レプリケートされたリポジトリを持つ実行インスタンスの1つが提供するリポジトリに基づいて動作します。これらのインスタンスは柔軟に使用させられるので、呼び出しパターンの変化に効率的に対応することができます。

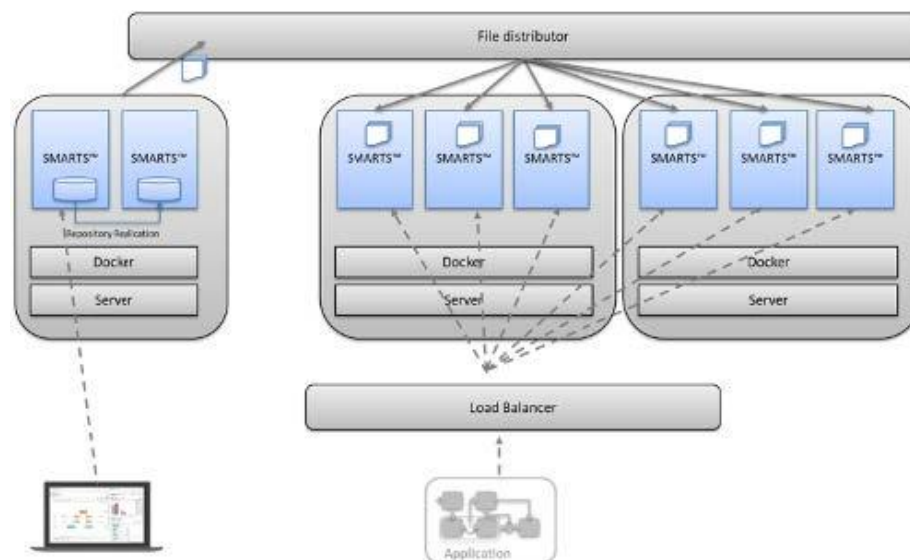


図 11 - リポジトリベースのデプロイ

このシナリオでは、リポジトリベースのデプロイメントと同じメリットを得ることができますが、以下の点が異なります：

- ☑ セキュリティは、SMARTS™ が直接ではなく、エコシステムによって処理されます。
- ☑ デプロイは SMARTS™ によって管理されるのではなく、ファイルをいつデプロイするかを決定するエコシステムによって管理されます。

また、IT企業・組織がオーケストレーションツールを活用して、残りの資産が配置されるのと同じ方法で SMARTS™ インスタンスを配置したいと思うかもしれません。

以下の例（図 12）では、前の例で説明したライトファイル配布メカニズムを、コンテナの完全なオーケストレーションデプロイに置き換える方法を示しています。

このアプローチでは、先に概説した利点を得ることができ、お気に入りのコンポーネントやオーケストレーションツールを使用して、セキュリティ、ログ管理などを完了させることができます。

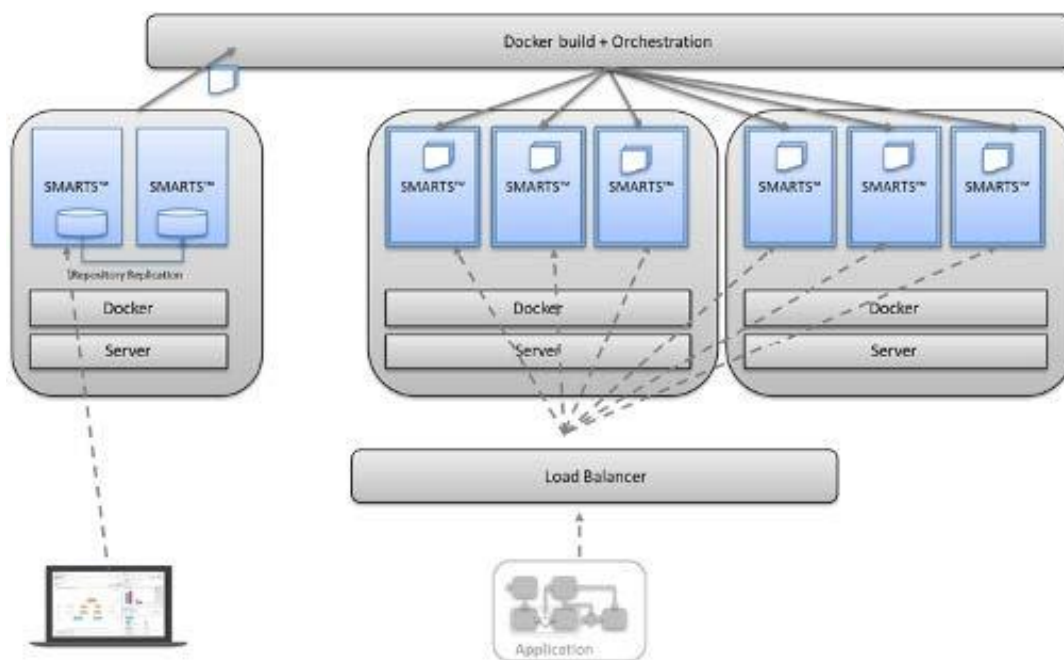


図 12 - デプロイにオーケストレーションツールを活用

## ● ● ● エコシステムへの適合

最後の章として、SMARTS™ の導入目的であるデジジョン管理を行うタスクが、いかに容易に実行できるかについて説明します。また、SMARTS™ をエコシステムに安全かつ効率的に統合するための開発およびテストについて考慮すべき点を説明いたします：

### セキュリティ

何よりもまず第一に、セキュリティに関する考慮事項があります。デジジョンは企業の中核的なナレッジであり、製品の操作の安全性を確保するための法的に保護された特別な仕様要件またはベストプラクティス（経験則、ノウハウ）のいずれかから構成されている場合があるからです。

### 認証

SMARTS™ 内で行うすべての操作は、有効な認証の状況下で行われます。SMARTS™ は組み込みの認証プロバイダを提供、この認証プロバイダは、資格情報を安全に(ハッシュ化されているなど)リポジトリに保存しますが、それ自体はお客様が管理する鍵で暗号化されています。一時的なパスワードは、ユーザーが登録されたときに作成され、電子メールを介してユーザーにのみ通知されます。

The image displays two overlapping screenshots of a software interface titled "Add authentication setup". The left screenshot shows the "Generic" protocol selected in a dropdown menu. The right screenshot shows the "OAuth 2" protocol selected. Both screenshots show a form with various fields for configuring an authentication setup, including Name, Relying party, Protocol, Authorization endpoint, Token endpoint, Metadata endpoint, User profile endpoint, Username expression, Unique ID expression, Client identifier, Client secret, Scope, and Additional options. The "Add" button is visible in the bottom right corner of both windows.

図 13 - 認証プロバイダと設定

しかし、独自の認証プロバイダを使用したいと思うこともあるでしょう。SMARTS™では、ユーザーがSMARTS™で認証しようとしたときに、認証要求が法人認証プロバイダによって実行され、法人が定義した設定とポリシーを使用することを保証する、委任認証プロバイダを設定することができます。オーサリング環境の場合、シングルサインオンアプローチの実装も可能になります。

SMARTS™ は、さまざまな認証プロバイダと設定、Microsoft AD LDS、Microsoft または Google Accounts、Genericなどをサポートしています。プロトコルには、LDAP、WS-Federation、OAuth2が含まれます。

## アクセス制御

認証されると、SMARTS™ ユーザーには、サインインしたアカウントに役割（ロール）が割り当てられます。ユーザーは1つ以上の役割を持つことができ、これらの役割によって実行可能な操作の種類が決まります。

以下の役割が事前に定義されています：

- ☑ **テナント管理者の役割**：この役割はテナントの管理者に与えられます。
- ☑ **テナント・ライフサイクルマネージャの役割**：この役割は、管理機能にはアクセスできませんが、ライフサイクル機能（タスクフロー、ライブラリのインポート、公開、デフォルトリリース管理）にはアクセスできるユーザーに与えられます。
- ☑ **ワークスペース管理者の役割**：この役割は、アカウントを持っているワークスペースのユーザー アカウントを編集できるユーザーに与えられます。
- ☑ **ワークスペース・ライフサイクルマネージャの役割**：ユーザーがライフサイクルマネージャであるワークスペース全体のライフサイクルを管理します。
- ☑ **リリース管理者の役割**：リリースの公開とデフォルトリリースの管理のみが許可されます。
- ☑ **ユーザーの役割**：これは、ワークスペースのユーザーのためのデフォルトな役割です。これは SMARTS™ Decision Management 機能を使用するユーザーの役割となります。
- ☑ **制限されたユーザーの役割**：デシジョンロジックの更新は許可されていますが、デシジョンフローやフォーム構造の変更は許可されていません。
- ☑ **ビジネスページユーザーの役割**：ビジネスハブを通じて、事前に作成されたビジネスページとのアクセスを可能とする対話を許可します。

さらに、お客様は、ユーザーインターフェースである「コマンドセット（500以上）」にアクセスするだけでなく、権限を指定して独自の役割を追加することができます。

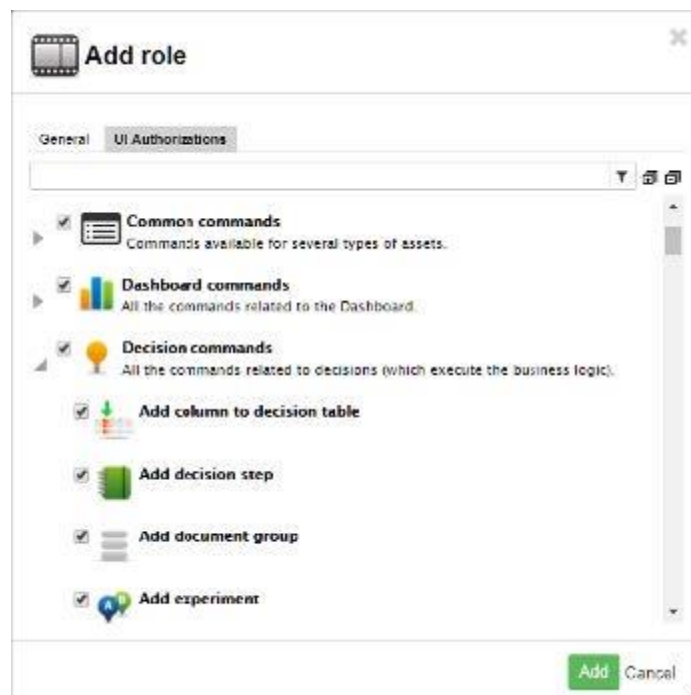


図 14 - 新たな役割のアサイン

## プログラム操作のセキュリティ

すべてのサービス・プログラム上の操作は、安全な認証の状況下で、しかもアクセス可能な役割によって行われます。このような相互操作はすべて HTTPS を通して実装されております。

さらに、すべての API 呼び出しは、アプリケーション ID（各ユーザに個別に割り当てられた）と秘密鍵（顧客が管理する）を組み合わせることで計算されたハッシュによって保護されています。このハッシュにより、SMARTS™ は、呼び出しのペイロードが改ざんされていないことを確認することができます。

ISO 形式のタイムスタンプも、インフラストラクチャの残りの部分で処理されない場合の繰り返し行われる攻撃から保護するために使用されます。

## ストレージのセキュリティ

データベースとのすべてのやりとりは SMARTS™ の内部で行われ、HTTPS を通して行われます。さらに、お客様は自分のコントロール下で暗号化キーを指定することができ、データベースの内容を暗号化するために使用されます。

## トレーサビリティ

デシジョンは組織にとって重要な資産であるため、厳重に管理する必要があります。SMARTS™ は、すでにカバーしているバージョンングとセキュリティに加えて、リポジトリに対して実行された操作のすべてをトレースする完全なアクティビティストリーム(図15参照)を追加しました。トレースでは、誰が、いつ、どこで、何をしたのかの証跡をとることができます。

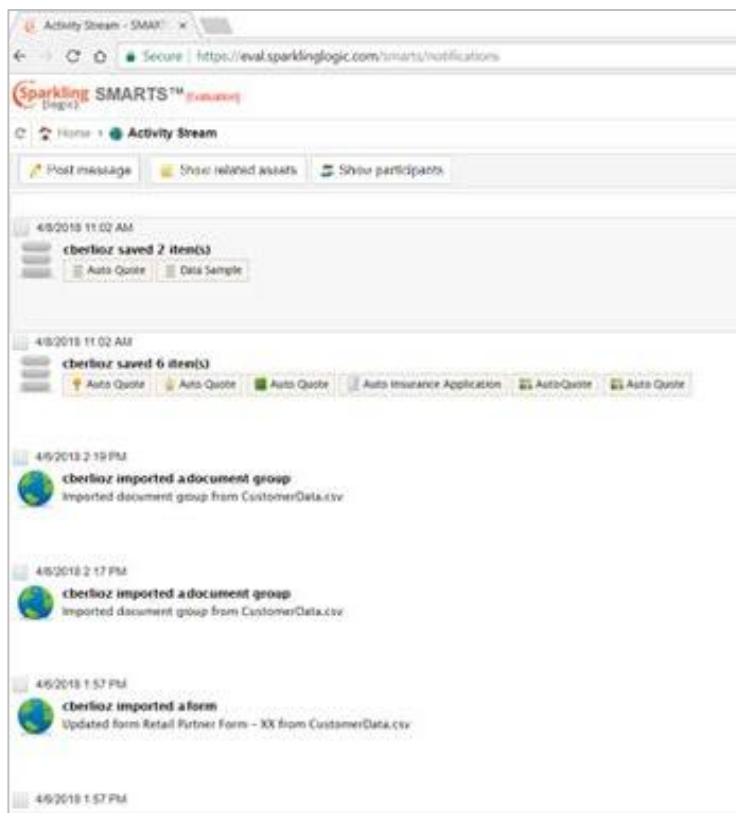


図 15 -アクティビティ・ストリーム

アクティビティ・ストリームは、プロフィールを介して設定された電子メールを通して加入者に伝達することができます。さらに、アクティビティ・ストリームは外部のエンタープライズ通知システムにルーティングされ、保存されたり、他のエンタープライズアセットを作成したり、アラームを発生させるために使用することができます。例えば、お客様は アクティビティ・ストリームの特定の操作を、Atracian's Jira ソフトウェア (プロジェクト管理ツール) 使用のシステムでチケット起票に結びつけて活用しています。このような統合は、Sparkling Logicで定義されているインターフェースに従って、お客様がRESTサービスを実装しSMARTS™ 製品への登録として実現されています。

## ステータスチェック&ロギング

SMARTS™ インスタンスが動作すると、製品は、その動作に関連するロギングメカニズム情報にログを記録します。この記録には以下のものが含まれます（但し、これらに限定されません）。

- システムの状態と問題
- 管理・ライフサイクル業務
- デプロイの初期化と更新
- エラーと例外

コンテナを介して実装された SMARTS™ インスタンスの場合、このログはアプリケーション監視の目的で企業のロギングシステムと連携できます。

さらに、SMARTS™ は、そのステータスを確認するための操作機能を提供します。

- ☑ ヘルスチェックプローブ - 通常、特にデプロイメントマネージャからのデプロイメントの状態をチェックするために使用されます。
- ☑ システムステータスチェック - サーバーコンポーネントの基本的なステータスを返す http やりとりを素早く実行します。
- ☑ Rest デプロイメントのステータスチェック - すべてのデプロイメント・ステータスの詳細を返す認証REST からの呼び出しを実行します。

管理スクリプトまたはユーザーインターフェース - システム、リポジトリ、ディスクなどのステータスの詳細を提供します。

## データアクセス

前述したように、SMARTS™ がデシジョンを開発、テスト、最適化を行う際にはデータアクセスが中心となります。SMARTS™は、データアクセスの面について、以下のようなサポートを提供します。

- ➔ 複数のデータフォーマット
  - 通常、データはJSONまたはCSVで利用できますが、XML、Avro、PSVもサポートされています。
- ➔ 複数のデータソース
  - データはSMARTS™にアップロードすることができます - これは、使用中または開発中のデータ（ユースケース）、PII（個人を特定できない情報）を含まないデータ、および他で既に管理されていないデータに限定することができます。
  - データは外部ソースからSMARTS™にストリーミングすることができます - 一般的には ftp/ftps、sftp、http/https、Amazon S3などです。別のデータソースがある場合は、SMARTS™ で定義されたインターフェイスに沿って REST ゲートウェイを実装することができます。



しかし、実行時には、SMARTS™ のドキュメントまたはストリーミングインターフェースを介してデータを SMARTS™ に渡す必要があります。SMARTS™ は、リモートサービスアクセスを使用しない限り、データを引き出すことはありません。

## リモートサービス・アクセス

SMARTS™ のデシジョンを、エコシステムですでに利用可能なサービスと相互連携させたい場合があるかもしれません。これはデシジョン管理の逆のパターンと考えられますが、理にかなっている場合もあります。例えば：

- ☑ 小さいながらも重要な結論を出すために取得したいデータがあるかもしれません。呼び出しの割合-DMVレコードを取得して改善することを想像してください。それ以外の場合、境界線ぎりぎりであるクレジットアプリケーションのリスクプロファイルみたいな例
- ☑ デシジョンを継続するために、どのデータを取得するかを決定するデシジョンロジックがあるかもしれません。
- ☑ 予測モデルやスコアリングモデルがすでに実装されていて、サービスとして公開されていて、デシジョンに活用したい場合があるかもしれません。

SMARTS™ はリモート機能をサポートしています。リモート機能は IT 部門によって設定され、ビジネスユーザーが JSON-RPC で実装されたサービスや、ある種の REST をデシジョンロジックから呼び出すようなアクセスとして提供されます。

例えば、図 16 に示されているリモート機能は、ローカルの JSON-RPC ゲートウェイを介してアクセスされたリモートサービス（ここでは Zillow のユースケースを提供）を呼び出す機能をビジネスユーザーに提供し、物件の価格を決定する事例として使用されています。

The image shows two screenshots of the SMARTS™ configuration interface for a function named 'RetrieveAddressValueEstimate'.

The top screenshot shows the 'Signature' tab. It displays the return type as 'Address Value Estimate' and 'Just one value'. Under the 'Parameters' section, there are two parameters: 'Address' (String, Just one value) and 'ZipCode' (String, Just one value).

The bottom screenshot shows the 'Implementation' tab. It displays the 'Kind' as 'Remote'. The 'Service URI' is 'http://localhost:8964/jsonrpcserver.rpc' and the 'Remote name' is 'RetrieveAddressValueEstimate'.

図 16 - リモート機能

SMARTS™ 製品は、サーバー側のメッセージ(ログなど)とユーザーインターフェースの両方で、国際化されています。現在、以下のロケールにローカライズされています：

英語、スペイン語、簡易中国語、日本語

SMARTS™は、契約の一部としてブランド化することができます。ブランディングは、すべての会社名や製品名、資産参照（URL、電子メールアドレス、コンテンツ）を、お客様から提供されたものに置き換えます。オンラインドキュメントもブランド化することができます。

まとめ

SMARTS™ Decision Managerは、デシジョンの発見からデプロイまでのDecision Management ライフサイクル全体をサポートするエンタープライズ対応のDecision Managementプラットフォームです。このプラットフォームは、統合、デプロイ、アーキテクチャの面で想定される問題を最小限に抑えながら、多種多様な顧客環境に対応できるように設計・開発されています。

このホワイトペーパーでは、アーキテクト、開発者、その他のIT専門家の関心事や懸念事項に対処するために、製品のより技術的な側面に焦点を当てました。ライフサイクル管理、デプロイメント、オプション、パフォーマンスとスケーラビリティ、統合などについて記載しております。

ご質問や詳細な情報につきましては、お気軽にお問い合わせください。

### ■お問い合わせ先■

Sparkling Logic SMARTSの製品説明、デモ/無償トライアル、  
Digital Decisionの基礎トレーニングにつきまして、  
下記にお問い合わせください：

[info@digital-decisioning.com](mailto:info@digital-decisioning.com)

- Sparkling Logic Incは、カリフォルニアベースの会社で、組織の「現場におけるデジタル・デシジョンング」を競争優位に展開できる「Decision Management」のテクノロジーカンパニーです。主要製品のSMARTS Decision Managerは、マルチ・ルールエンジン、分析プラットフォーム、機械学習を装備し、データインフォームド・デシジョン（データをひとつの情報としたデシジョン）のモデリング、開発、テスト、シミュレーション、継続的な改善、予測モデルによる最適化 等が容易に行えるシステム開発の『**行動を決定するDecision Management Suite**』ツールです。
- The Team, Digital Decisioningは、Rules Managementを超えた「Decision Management」テクノロジーをベースに、デジタル時代のデータトランスフォーメーション（DX）をめざすお客様むけに、Digital Decisioningのテクノロジーサービスを提供する専門家チームアップされた合同会社です。  
Digital Decision Managementについての基礎教育、モデル化支援、Decision Management導入コンサルティング、システム導入支援等を「ルールベースAIと機械学習AIのハイブリッドソリューション」でサービスするチームです。

□◆□詳細は、[www.digital-decisioning.com](http://www.digital-decisioning.com)をご覧ください