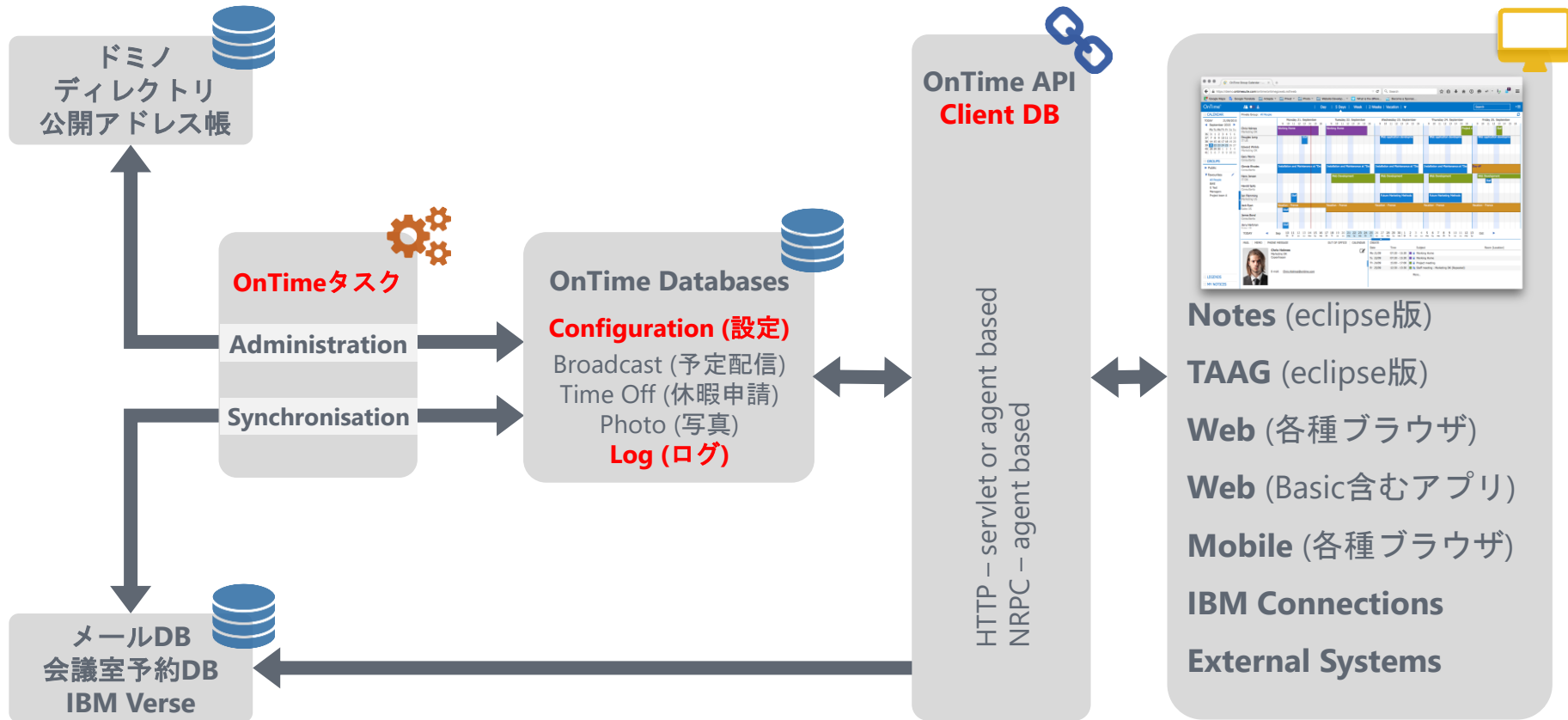


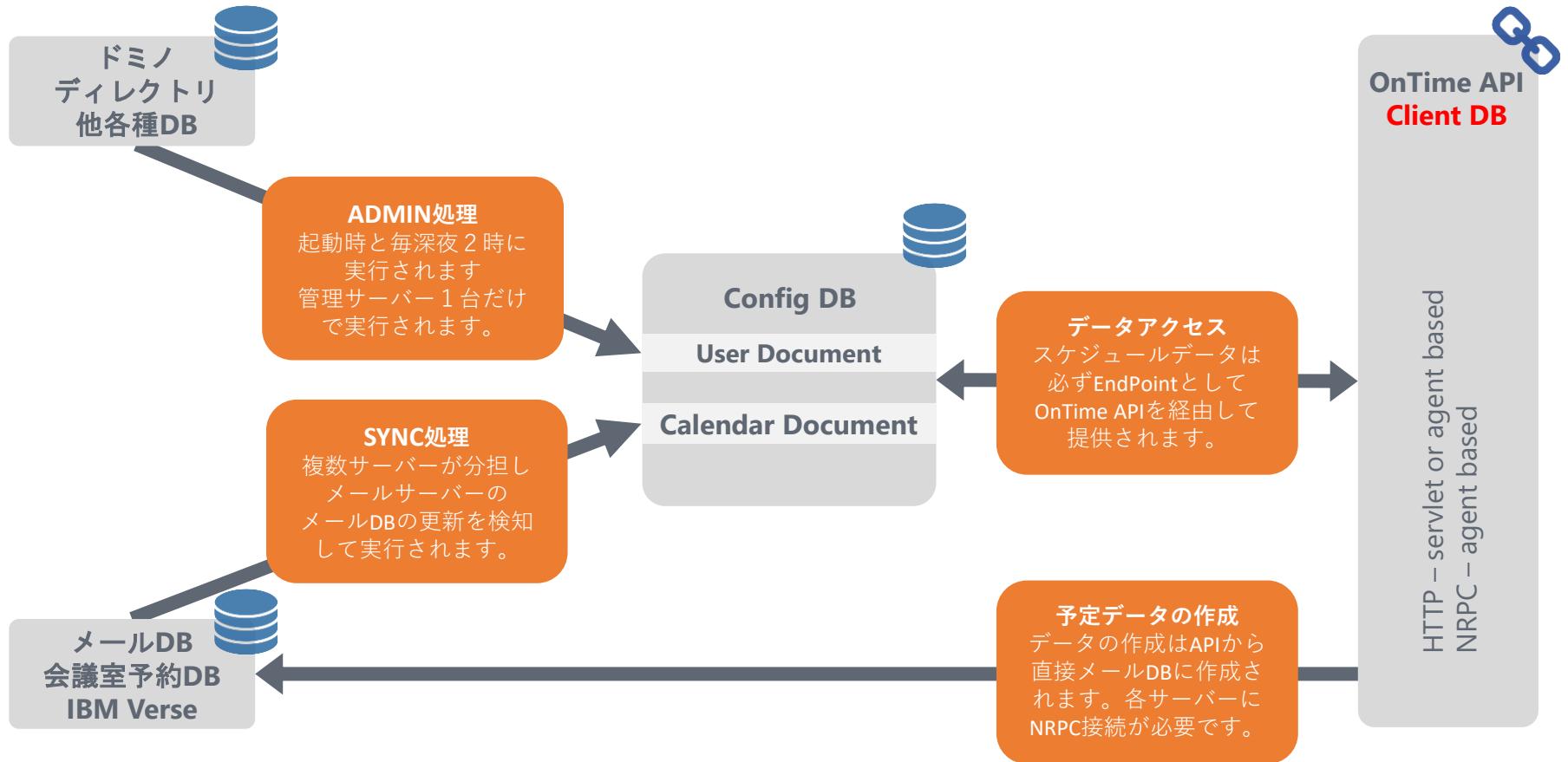
OnTime for IBM

技術資料(暫定版)

1.OnTime for IBMの構成



2.OnTimeの 4 つの主要な機能



1.ConfigDBを構成する文書(手動作成)

Global Settings

- 導入するOnTime環境に1つだけ作成する文書。全体設定に関連する情報は全てこの文書に集約されている。
- 参照するディレクトリ、同期対象のユーザーやグループ、同期期間、参照するプロフィール情報、ソート条件、等。

Server Settings

- 導入するOnTimeサーバー毎に作成する文書。そのサーバーがモニターするメールサーバーの指定・スレッド数等のタスク関連と、クライアントDBのURL指定・サブレットの使用等のAPI動作に必要な情報を記述。

Roles

- ユーザー・グループ単位で、実行できる権限レベル別に相手先のメールDBの所有者を登録する。
- デフォルトでNotes/Dominoと同じ設定が1つだけ準備。随時追加変更が可能。

Name Formats

- メインビューやプロフィールビュー用の名前や部署情報などの表示形式を設定する。
- デフォルトで英語系フォーマットの設定が3つだけ準備。通常は日本語表示用を作成する。

Default Settings

- ユーザー・グループ単位で、各種設定を行います。使用言語や表示形式、EnterpriseScaling等を指定可能。
- デフォルトで1つだけ準備。帯域の細かい拠点やユーザー表記を複数個運用出来ます。

Custom Fields

- Notes標準以外のフィールドを追加します。値はユーザーはメールDBに、リソースの場合はConfigに保持されます。
- 保存先毎に使用する/しないを指定することが可能。

Groups Static

- ドミノディレクトリのグループ文書と同じように個別にグループを作成します。
- このグループを利用出来る利用者を制限することが出来ます。

Groups Dynamic

- ノーツビューの1列目に使用するようなノーツ式でグループを自動生成・更新出来、メンテフリーを実現します。
- ドミノディレクトリ以外もソースとして利用出来るため、リソース予約DBや社員情報DB等も活用できます。

Groups Directory

- ドミノディレクトリのグループを条件設定に基づいてコピーして運用出来ます。
- 但し、コラボ基盤と同様英数字以外の文字で作成されたグループ文書は利用出来ません。

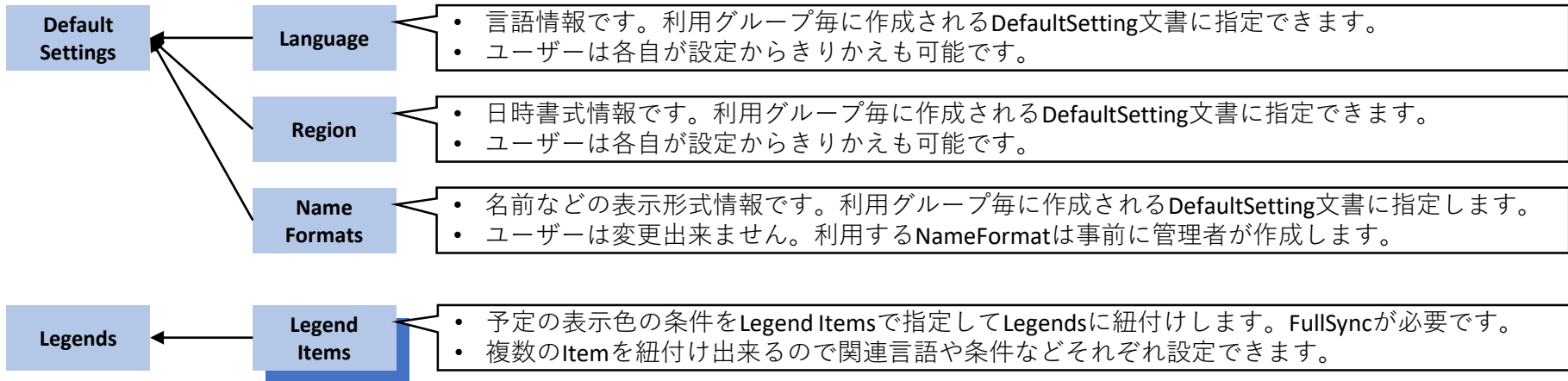
Groups External

- 全ての項目を独自に作成するDBで代用することが出来ます。
- 別DBなのでConfigのACLと別のアクセス権限で運用出来ます。但し、全ての項目を準備する必要があります。

Legends

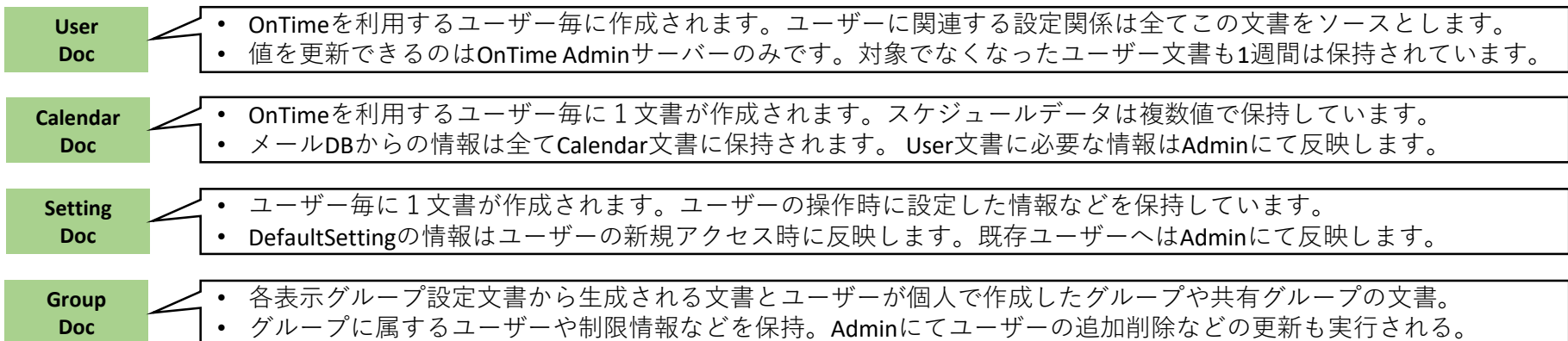
- 予定の表示色を指定します。各言語別の凡例表示名も準備出来ます。
- 表示条件はカテゴリだけでなく、ノーツ式を活用して指定できます。変更後更新にはFullSyncを必要とします。

1.ConfigDBを構成する文書（その2）



2.ConfigDBで自動生成される文書

以下の4文書は前述の設定文書群からAdmin処理とSync処理で自動生成されます。



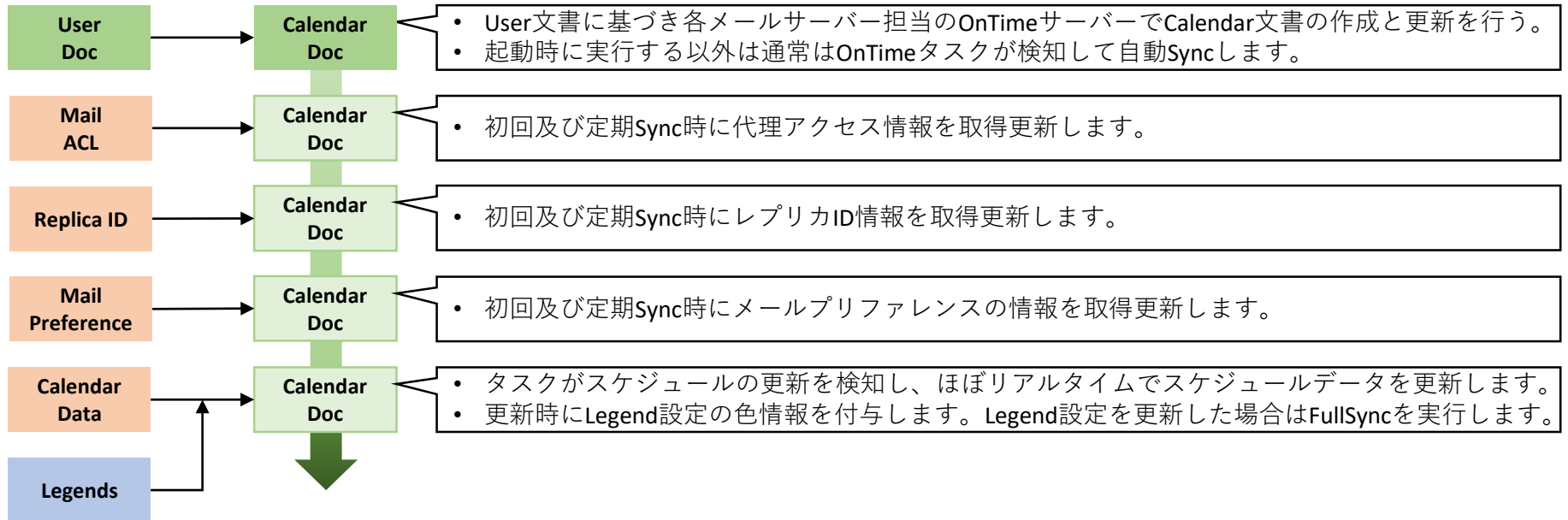
3.Admin処理（ユーザー情報更新）



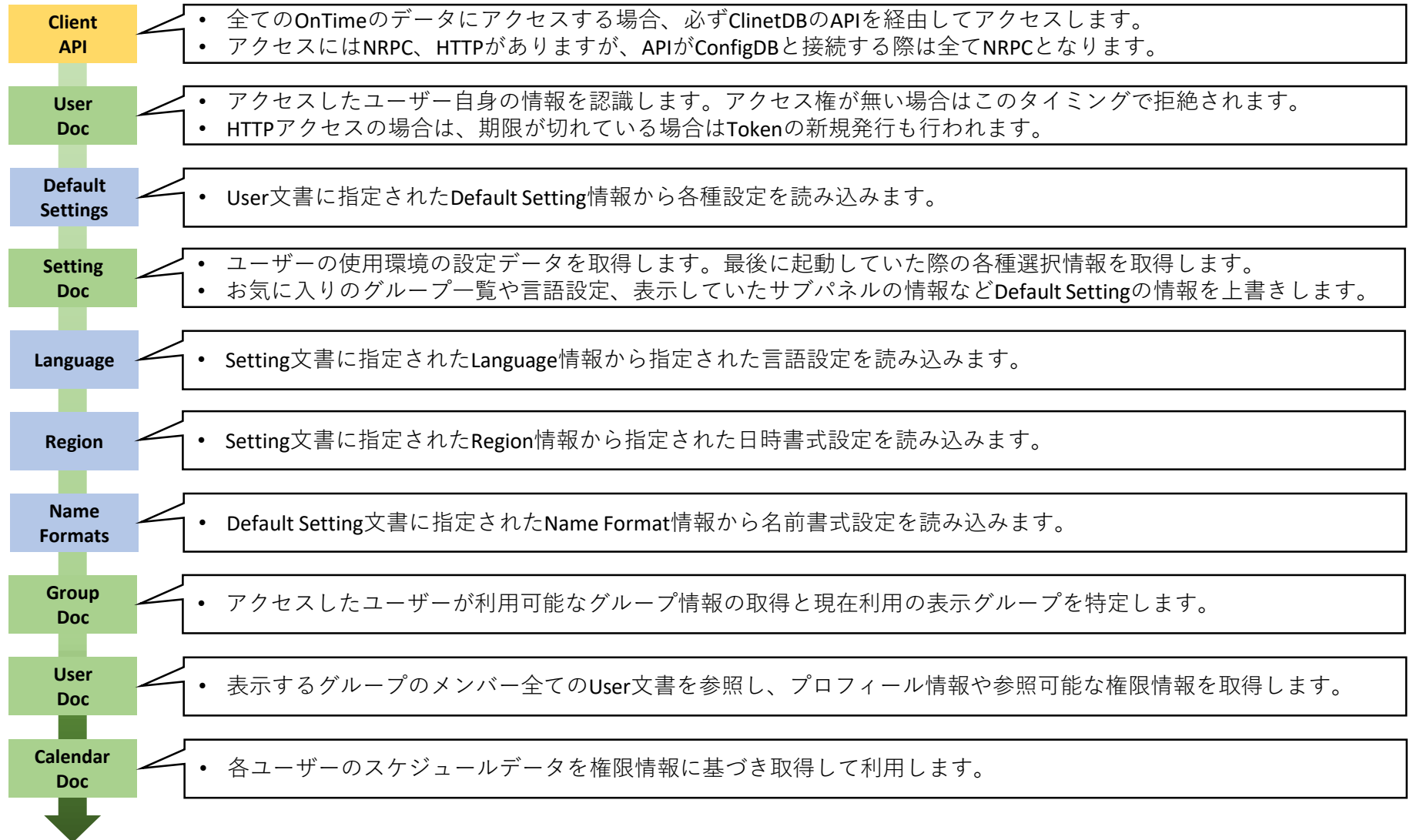
4.Admin処理（グループ情報更新）



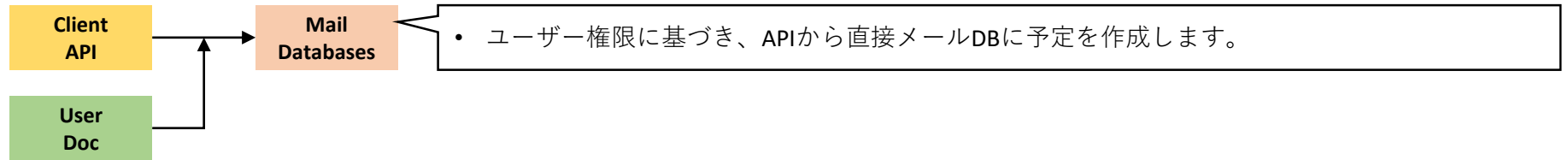
5 .Sync処理とFullSync処理



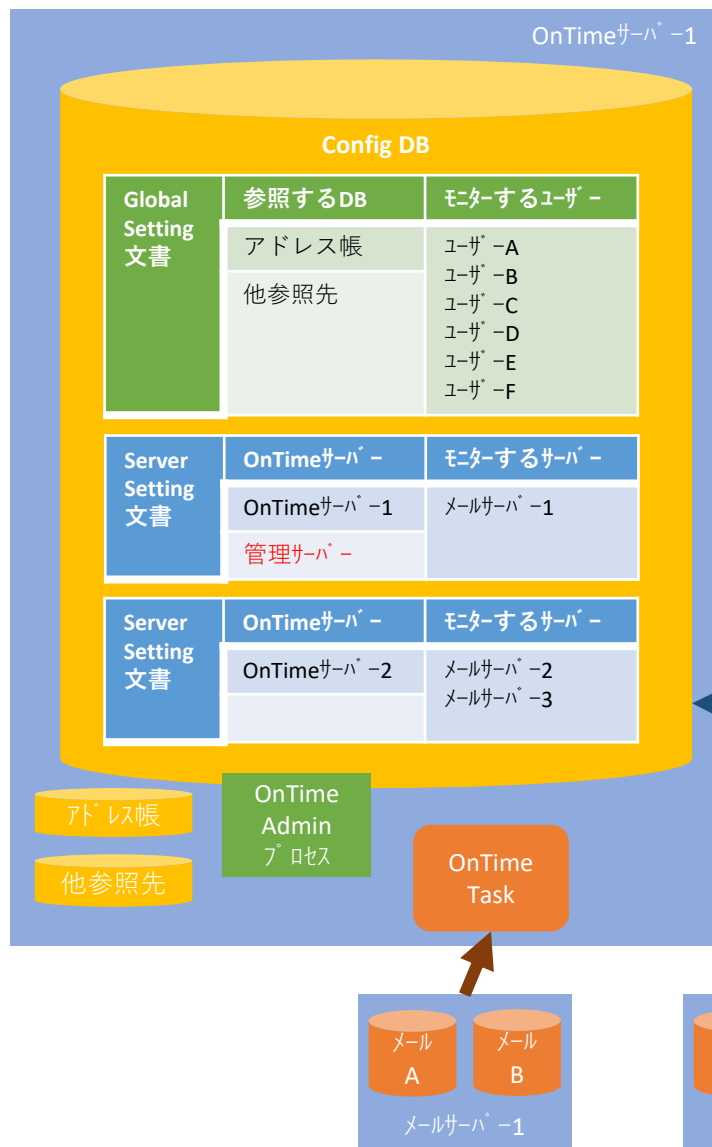
6. データアクセス



7. 予定データの作成や編集



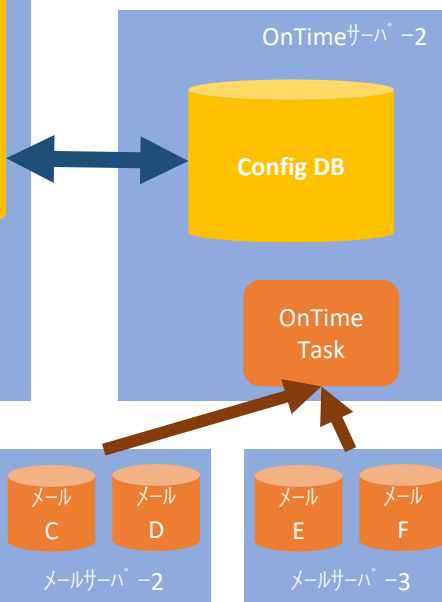
8.1.複製競合の回避ロジックについて



OnTimeはConfigDBに以下の情報を保持しています。

1. 各種設定情報（表示グループ、権限、凡例 他）
2. ユーザー情報「User文書」（1文書/ユーザー）
3. スケジュール情報「Calendar文書」（1文書/ユーザー）
（スケジュールデータは1文書内で複数值で保持）

複数のOnTimeサーバで構成する場合は、ConfigDBを複製して全てのサーバで同一データを保持することでスケラビリティを確保しています。



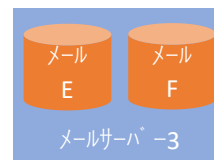
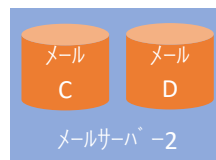
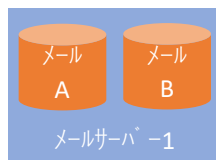
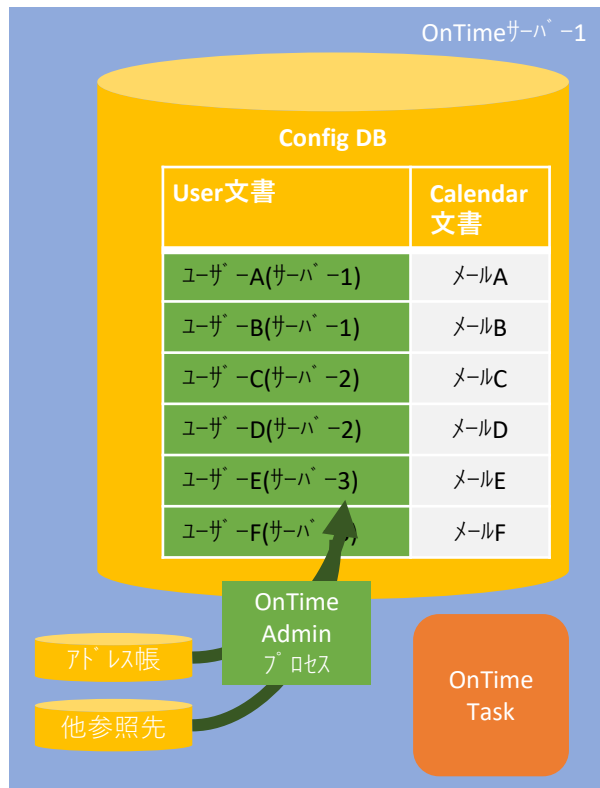
サンプル構成について

- OnTimeを2台のサーバに実装します。
- OnTimeサーバ-1はOnTimeAdminを兼務し、メールサーバ-1もモニターします。
- OnTimeサーバ-2はメールサーバ-2とメールサーバ-3の2台をモニターします。
- 利用するユーザーはユーザ-Aからユーザ-Fの6名とします。
- 通常OnTimeタスクはそれぞれのサーバで常時実行されています。

それぞれの情報が

- GlobalSetting文書(1文書/構成)
- Serversetting文書(1文書/サーバ-)に設定されているとします。

8.2.Admin処理の実行



OnTimeGC Adminプロセスは構成内のOnTimeサーバ 1 台だけで実行される機能です。デフォルトでは起動時と深夜 2 時に実行されます。

主に以下のような処理を行います。

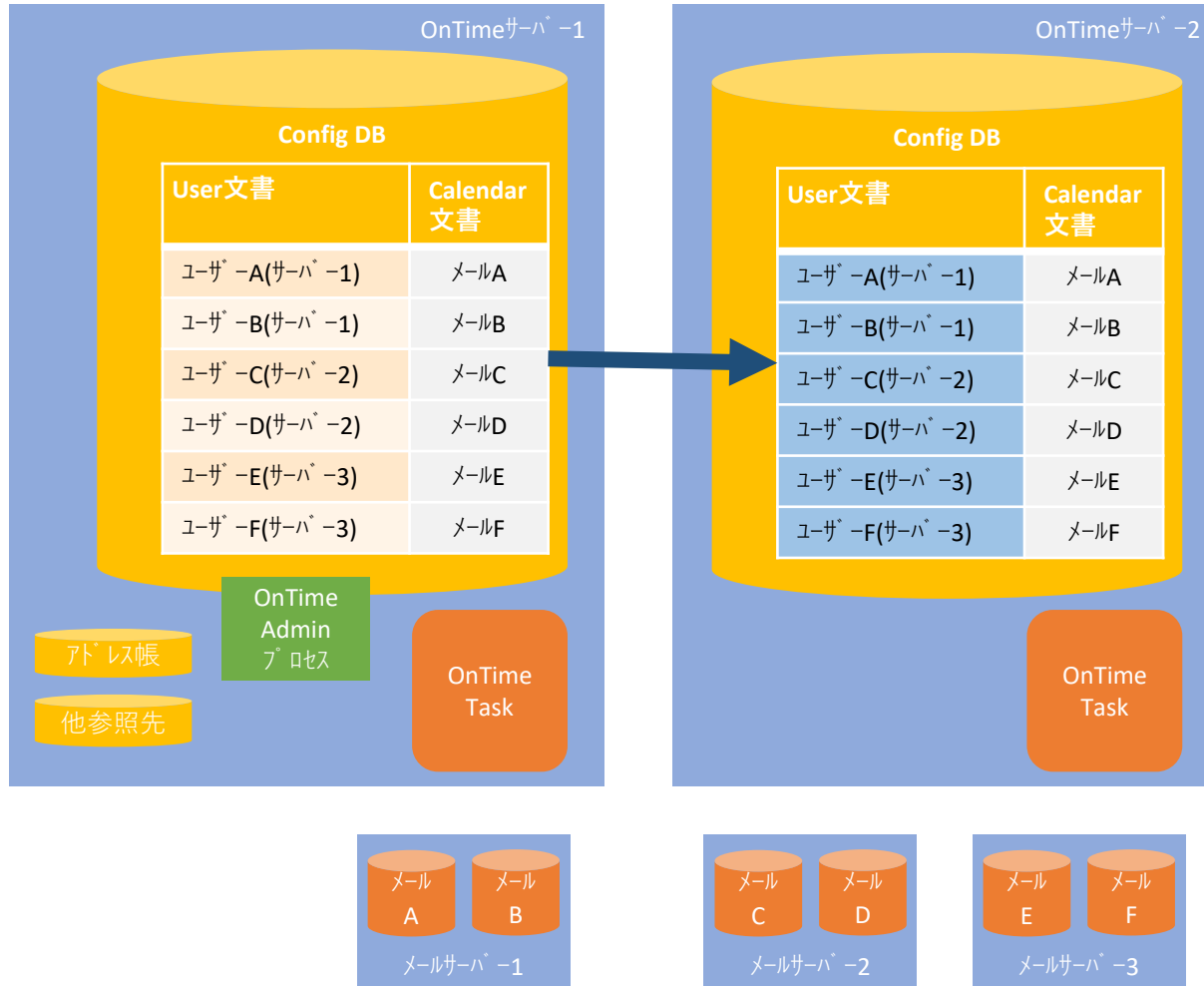
- 同期するユーザーのユーザー名やメールDBの情報をディレクトリから取得します。
- 同期するユーザーのその他プロフィール情報を参照指定されたDBから取得します。
- 作成されたUser文書にロール設定を行います。
- 作成されたUser文書からグループを生成します。

等々

左図ではUser文書が新規生成される状態を表示しています。同期するユーザーや会議室毎に 1 文書作成され、Adminプロセスの実行毎に更新されます。詳細は以下のリンクを参照。

<http://www2.ontimesuite.jp/moreinfoadmin/>

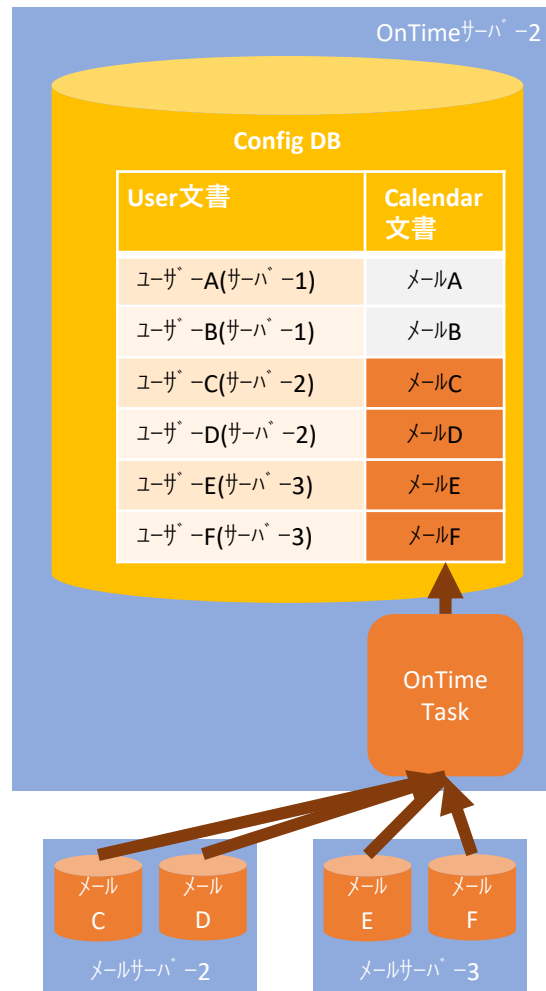
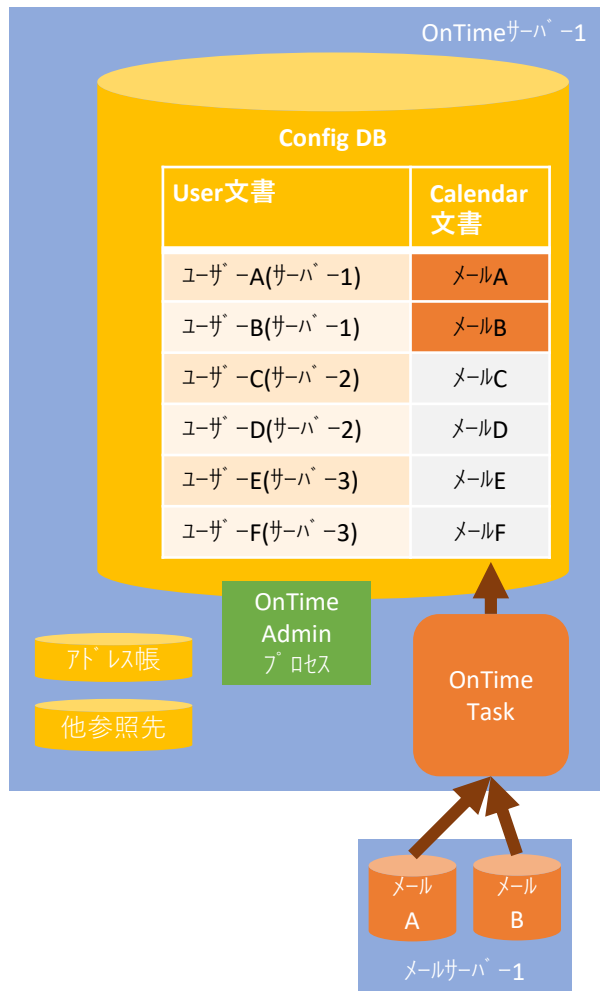
8.3.User文書を他サーバーに展開



作成及び更新されたUser文書を他のOnTimeサーバーのConfigDBに複製します。

複製はDominoの機能で設定します。OnTimeには複製に関する設定はありません。なので複製頻度やリポジトリはDominoディレクトリ内で設定して下さい。

8.4.Sync処理の実行



OnTimeGC Syncは各OnTimeサーバーのServerSetting文書に基づき、各サーバーのOnTimeタスクが実行します。

Syncはそれぞれモニターするサーバー上のメールDBや会議室予約DBを常に監視し、スケジュール情報の作成や更新が行われた際にリアルタイムで同期処理が実行されConfigDB内の複数值が保持されているデータフィールドを更新します。

また、SyncはメールDBが保持している以下のような情報も取得します。

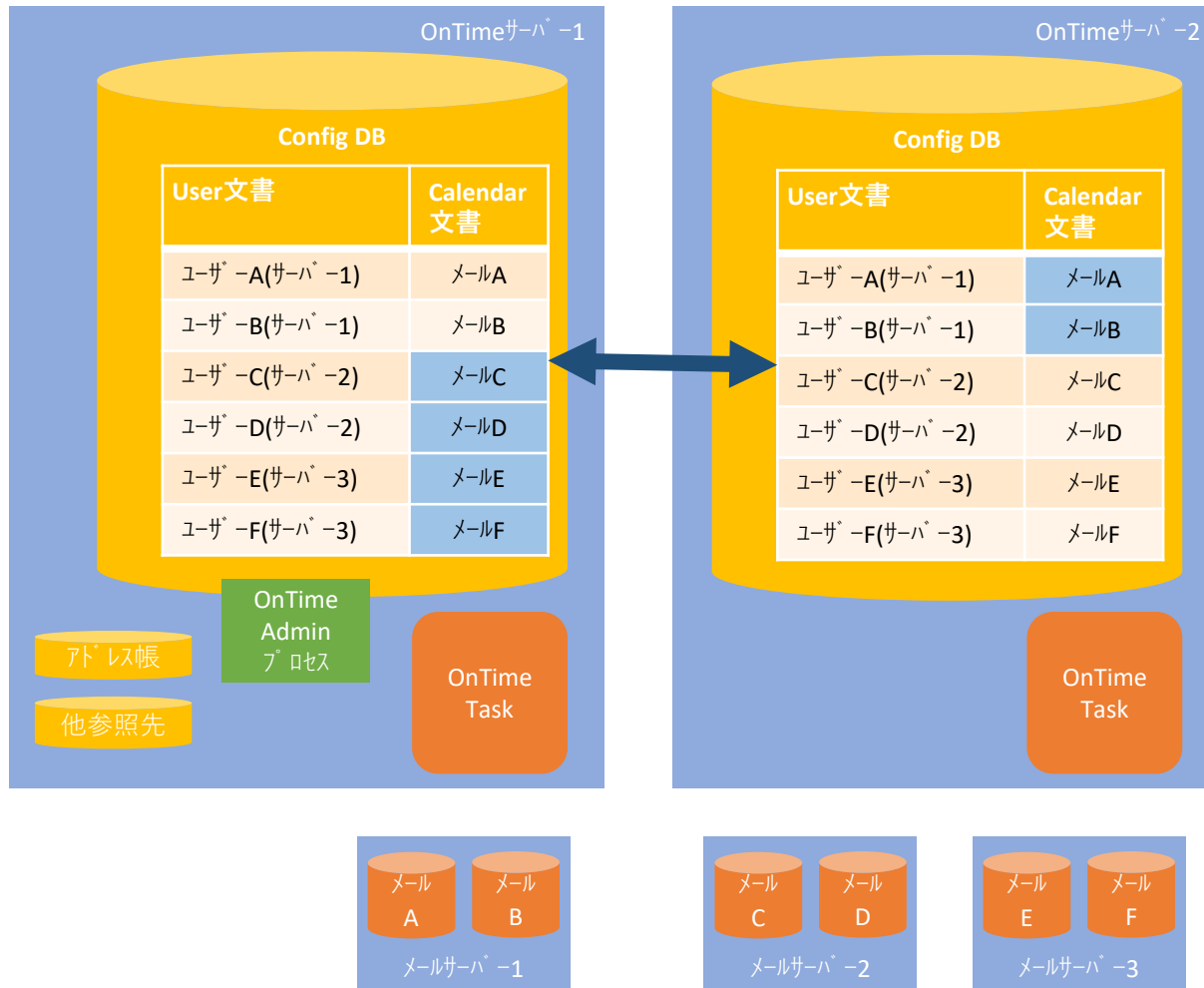
- メールのプリファレンス
- メールDBの代理アクセス権限
- 会議室予約DBのリソース文書上の各種制限事項
- レプリカID(フェールオーバー用)

初回起動時は全てのメールDBの処理が行われていないので、コマンドから強制的に実行します。

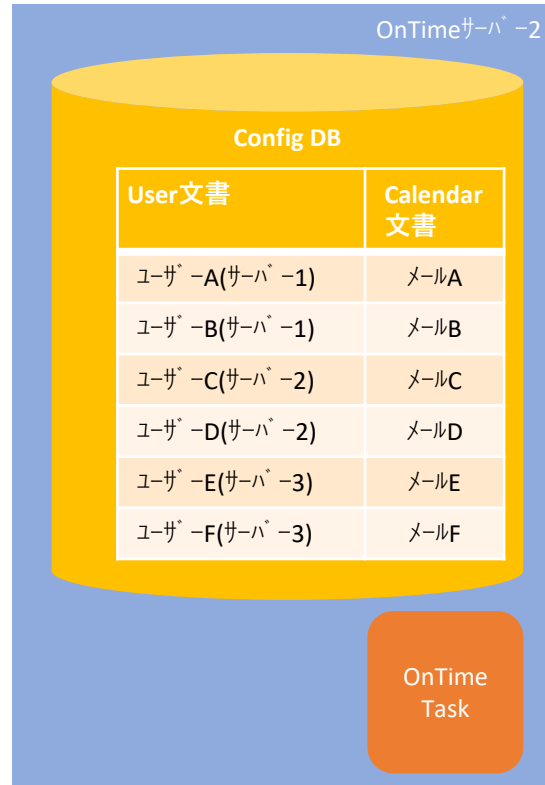
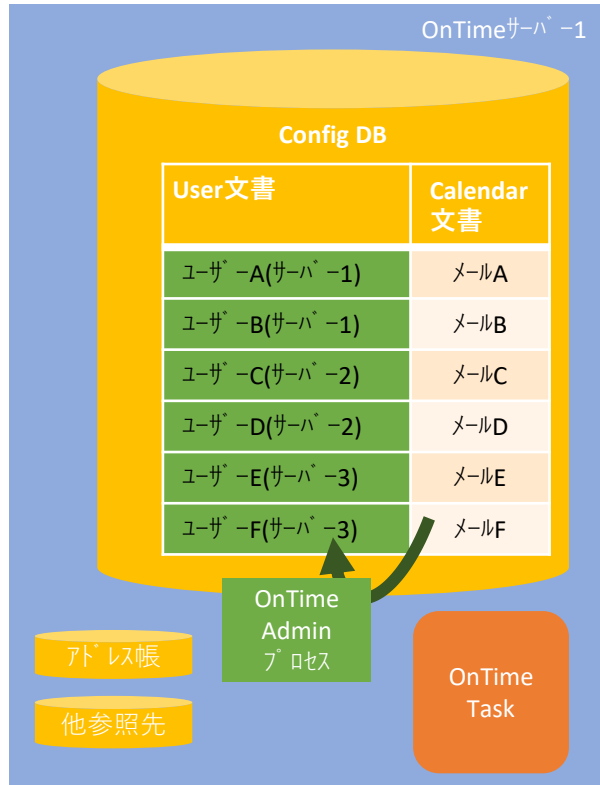
左図ではそれぞれのOnTimeサーバーが対象となるメールDBから対象期間のスケジュールデータを取得している状態を表示しています。

8.5.Calendar文書を他サーバーに展開

スケジュールデータや各DBが保持している各種情報を取得したCalendar文書もDominoの複製で管理サーバー含む他サーバーに複製します。



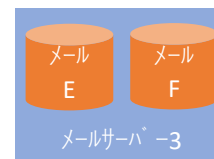
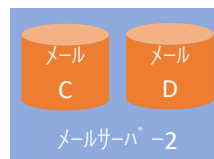
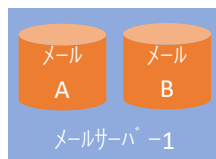
8.6.Calendar文書からUser文書に更新



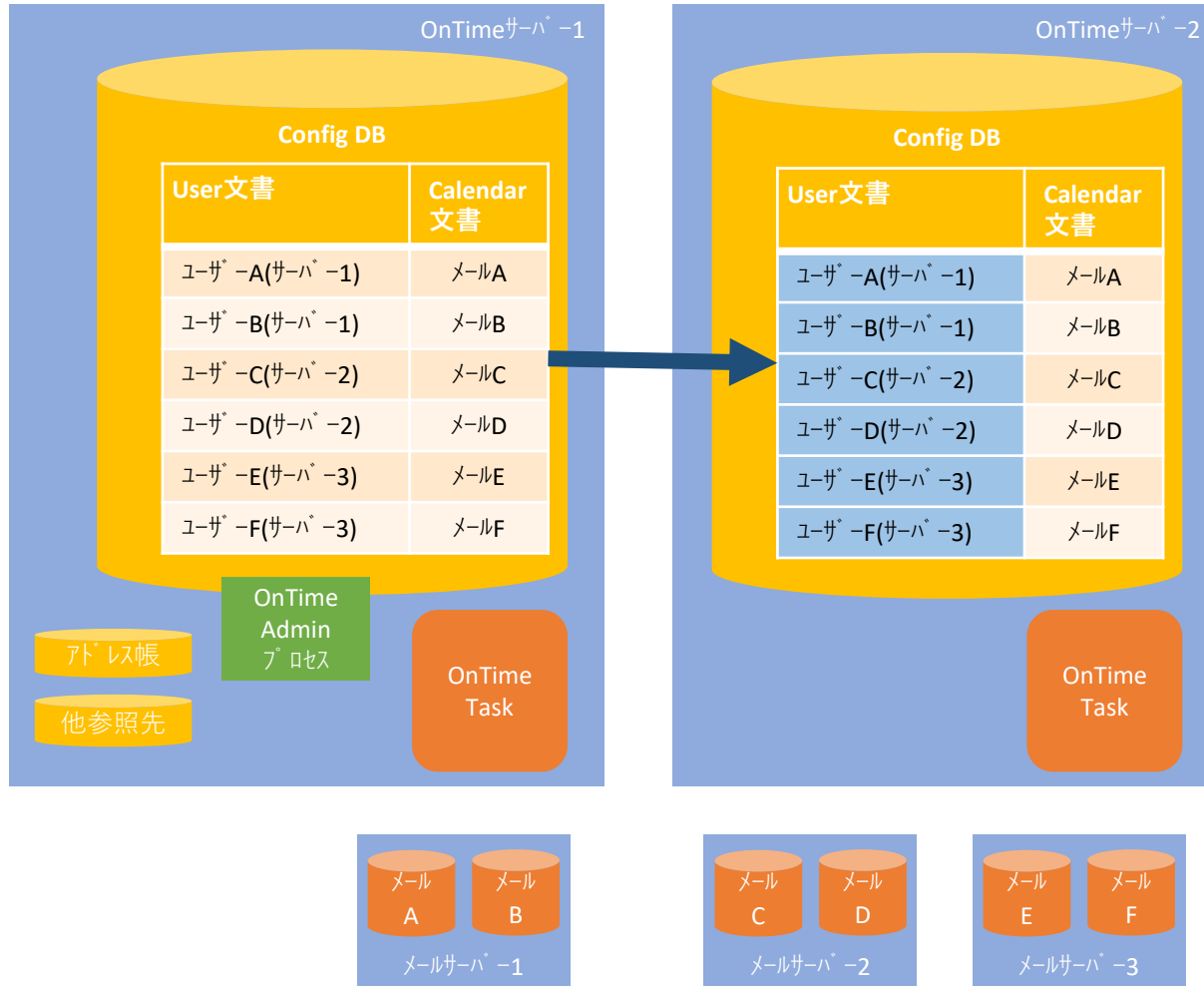
OnTimeの管理サーバーはOnTimeGC Admin処理で、ローカルにあるConfigDBに存在するCalendar文書内のスケジュールデータ以外の各種情報をUser文書に更新します。

Admin処理はローカル上のCalendar文書内の情報を使用して更新するので、複製が完了していない場合は複製前の情報で更新されますのでご注意ください。

ちなみに、OnTimeクライアントはスケジュールデータ以外はCalendar文書に同じ情報があっても必ずUser文書内の情報を参照します。



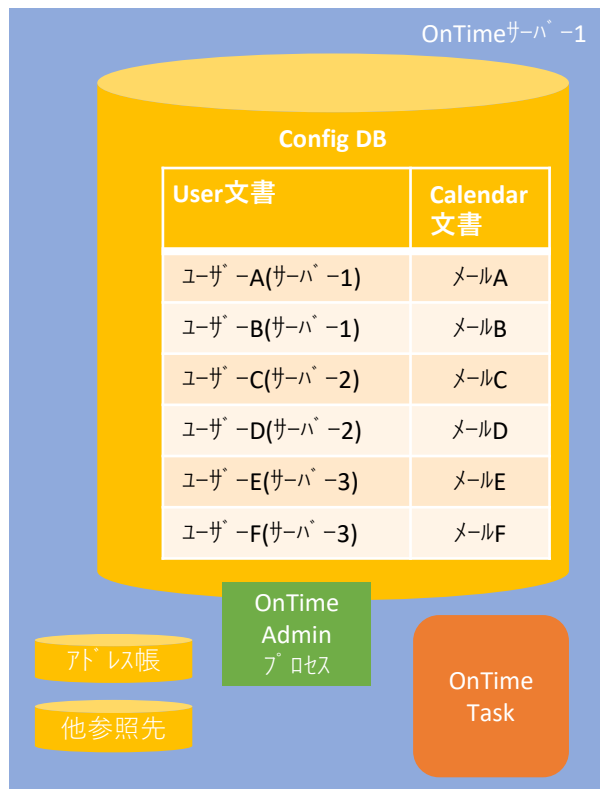
8.7.User文書を他サーバーに複製更新



更新されたUser文書を他のOnTimeサーバーのConfigDBに複製します。

複製はDominoの機能で設定します。OnTimeには複製に関する設定はありません。なので複製頻度やリポジトリはDominoディレクトリ内で設定して下さい。

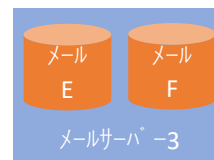
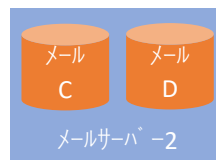
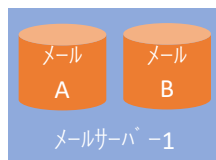
8.8.全サーバーで同一最新情報を維持



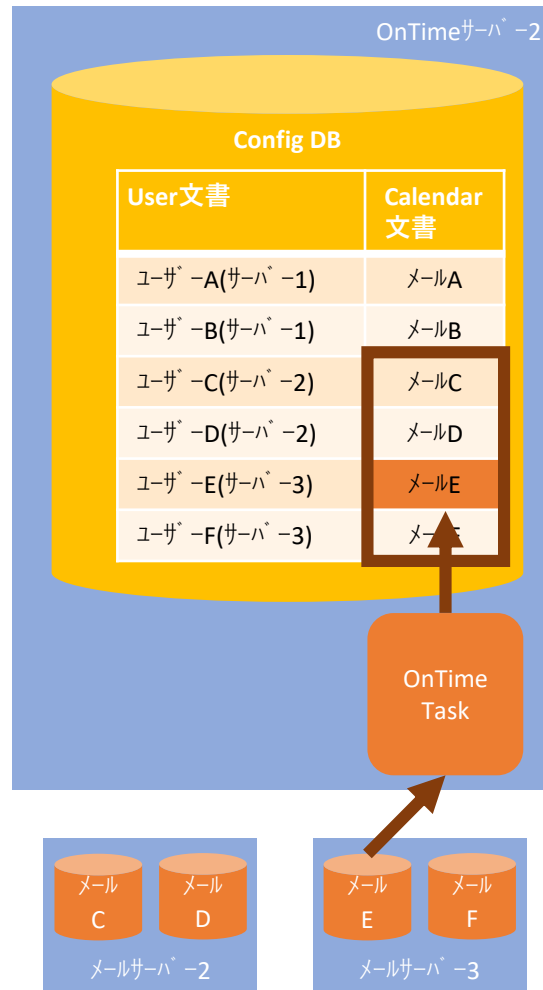
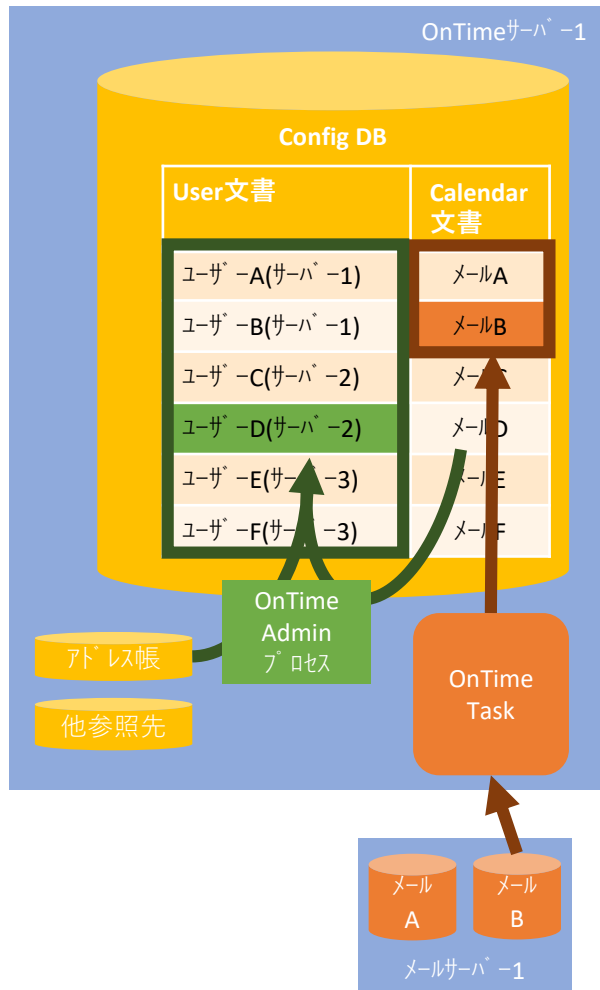
結果、全てのサーバーで同一の最新情報を維持出来ます。

後は自身の組織の特性

- 予定情報の更新頻度
 - ネットワークの負荷
 - ネットワークのトポロジー
 - サーバーのパフォーマンス
- に基づいてレプリケーションを構成してください。



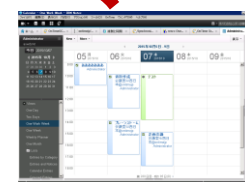
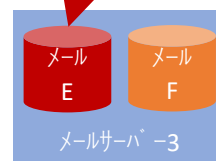
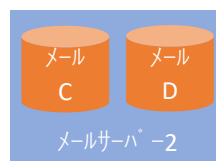
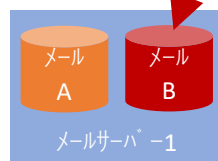
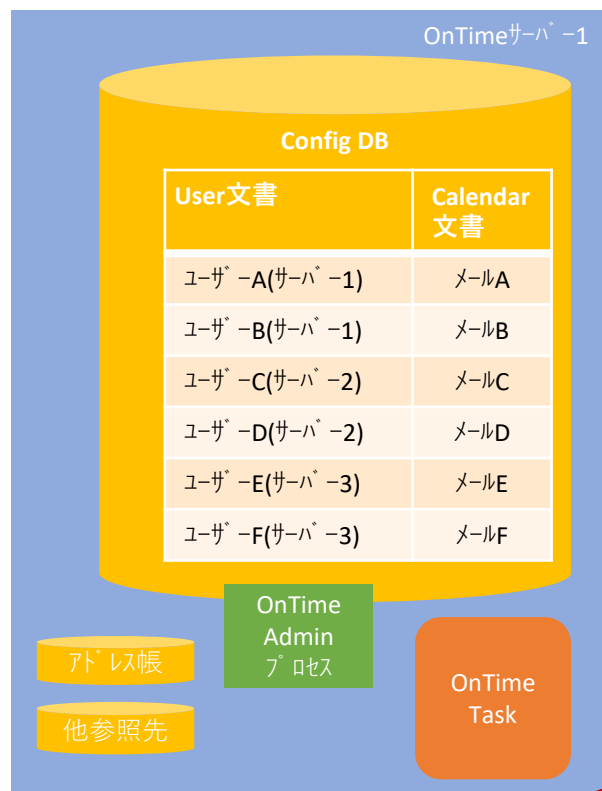
8.9.競合を回避するロジックの再確認



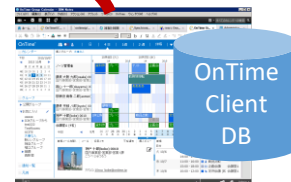
大きくは以下の3点によります。

1. 文書数が増えない。
例えばスケジュールデータが増えたとしてもConfig DB内の文書数は増えません。
結果、ビューのIndexerの負担は無く
例えば文書数が多いとしても文書特定のタイムラグは増えません。
2. 複数が1つの文書を更新しない。
1つの文書を保存出来るのは1つのロジックだけです。
Calendar文書を保存出来るのはそのメールDBを管理しているそのサーバーのタスクだけです。
User文書を保存出来るのはOnTime Adminプロセスだけです。
3. ユーザー操作によるスケジュールエントリ作成は直接メールDBに行われ、Config DBは関与しない。

8.10.予定は直接メールDBに作成・更新



メールDBから



OnTimeから

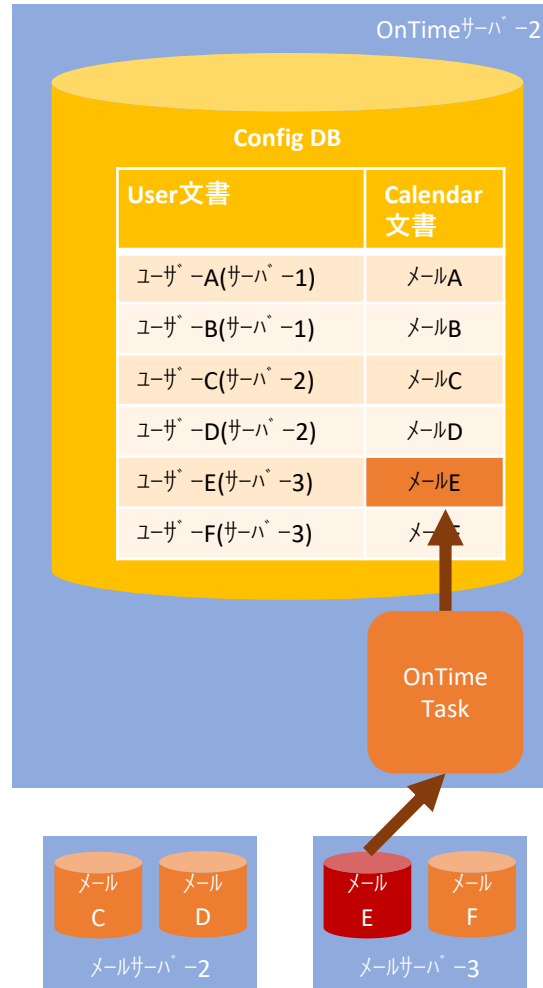
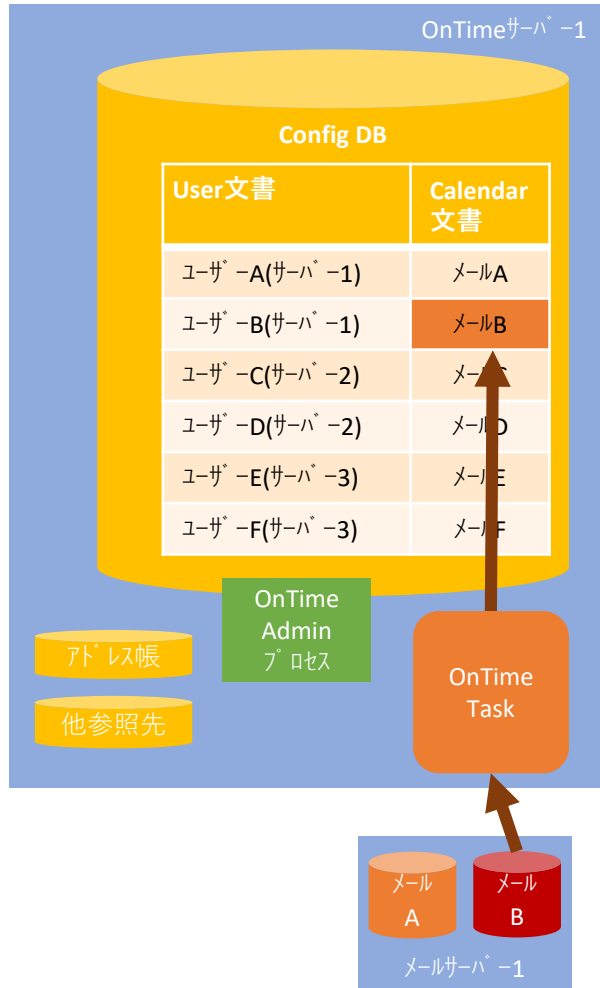
OnTimeは大きく2つの機能で構成されています。

1. メールDBからOnTimeへの同期機能(OnTimeGCタスクで動作)
2. メールDBへのスケジュール作成機能(Javaエージェントで実行)

ユーザーがメールDBに予定作成する場合は勿論、OnTimeから予定作成する場合は、OnTime Client DBのjavaエージェントが直接メールDBにエントリを作成します。ユーザー操作で直接Config DB内のデータを更新することはありません。

左図のようにメールDBに予定が作成・更新された場合は、、、次ページに続きます。

8.11. メールDB内で予定の更新を検知

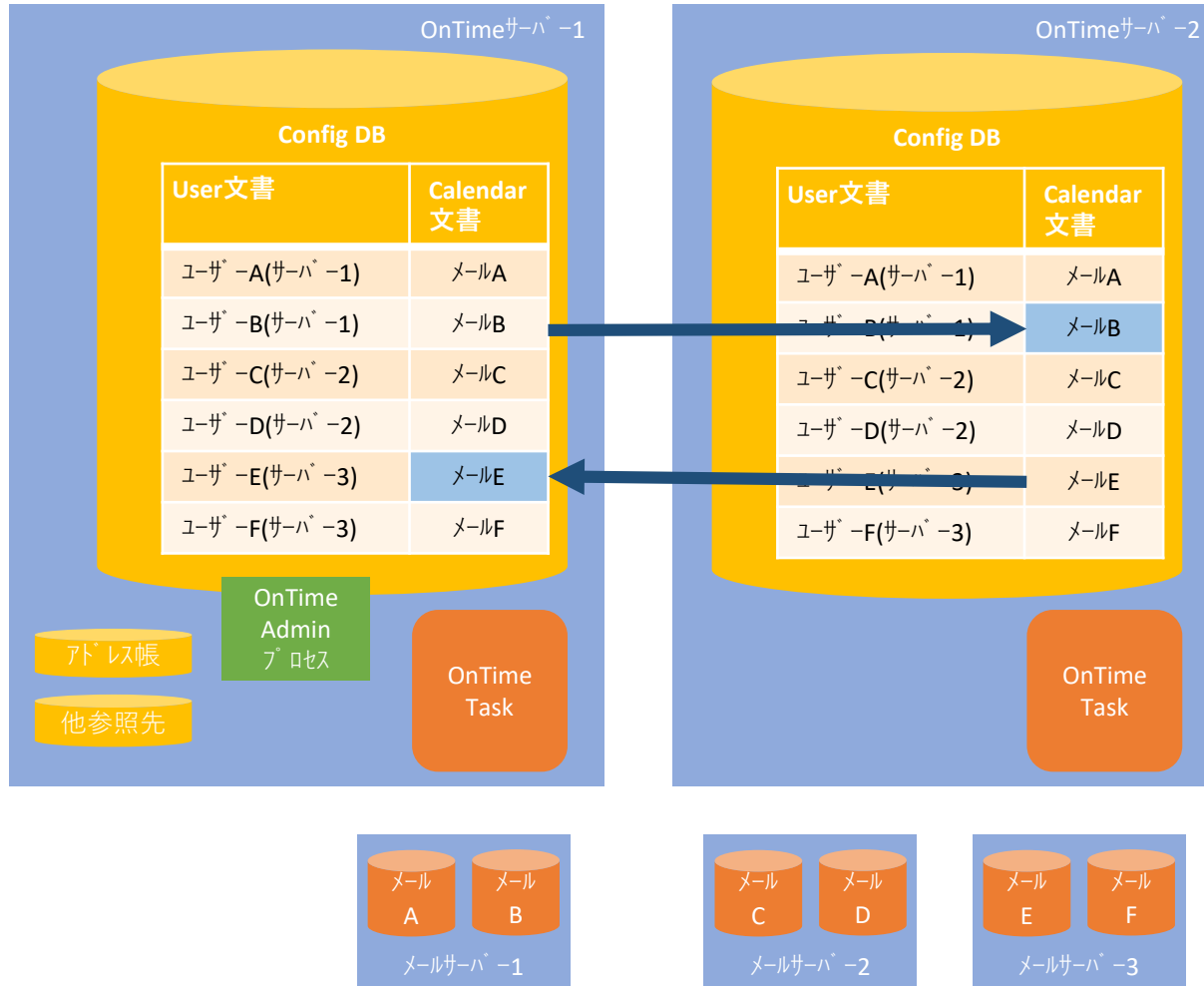


OnTimeタスクはメールサーバーと連携し、メールDB内の文書更新を検知します。

それはIBM Notes Travelerのそれと似たロジックと考えてください。

先ほど検知した文書更新はリアルタイムにタスクと同じサーバー上のConfig DBのCalendar文書内のデータを更新します。

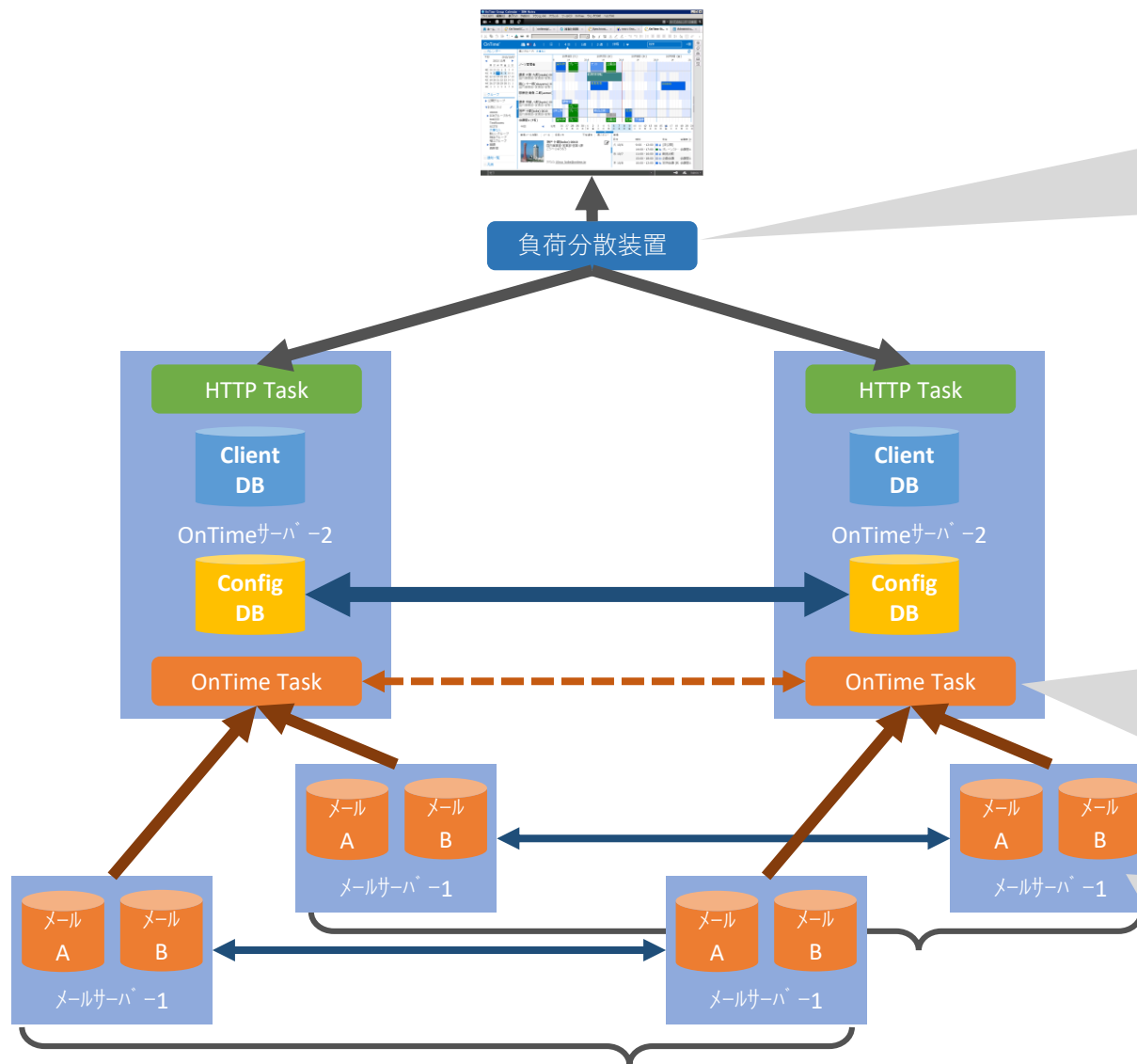
8.12.他OnTimeサーバーに情報更新



作成及び更新されたCalendar文書を他のOnTimeサーバーのConfigDBに複製します。

複製はDominoの機能で設定します。OnTimeには複製に関する設定はありません。なので複製頻度やリポジトリーはDominoディレクトリ内で設定して下さい。

9. フェールオーバーについて



クライアント向けフェールオーバー
HTTPの負荷分散装置等を導入して負荷分散、フェールオーバークラスタを構成して下さい。

ServerSetting文書のREST用のClientDBのURLには負荷分散装置のURLかアドレスを指定して下さい。
制御も負荷分散装置で行います。

負荷分散装置を使用しない場合
冗長化だけならエンドユーザーが手動でClientDBを開け直す、URLを接続し直す方法で切り替えることも可能。

OnTimeタスクのフェールオーバー
OnTimeタスクはServerSetting文書で指定したOnTimeクラスタメンバー間で死活監視をしています。
Dominoサーバーは正常でOnTimeタスクだけがダウンした場合、アクティブなクラスタメンバーが代理でダウンしたOnTimeタスク分もSync処理を司ります。

メールDBのフェールオーバー
OnTimeはUser文書にメールDBのクラスタパスを保持しています。何かしらの原因でメールDBにアクセスできなくなった場合、数回のトライアルの後フェールオーバー先に切り替わります。

10.OnTime Tokenについて

- OnTime Client DB(API)はDomino認証時に発行したOnTime Token(ユーザーを特定しその有効期間がエンコードされている暗号化文字列)でアクセスします。
- 一度Tokenが発行されると、OnTime APIへのRequest時に常に使用されOnTimeAPIとユーザーを特定する唯一の情報として利用されます。
- OnTime Group Calendarのセキュリティ機能(Tokenの利用)について
<http://www2.ontimesuite.jp/securityworks/>
- Ver.3.11.0以降のノートクライアントとサイドバークライアント(TAAG)の起動シーケンス
<http://www2.ontimesuite.jp/startup/>

11.クライアント操作のログトレース

- OnTimeのAPI通信は全てRESTによるRequestとResponseデータのやりとりになります。
- 特定のユーザーをDebugモードにすると内容を記録できます。
- クライアントがRequestを送信する際のコマンドやパラメータ及びデータサイズの調べ方
<http://www2.ontimesuite.jp/api-request/>
- サーバーとクライアントを流れるトラフィックデータについて
<http://www2.ontimesuite.jp/data-transfer/>

12. トラブルシュート（切り分け）

- [不在通知の設定が出来ない時のトラブルシュート](#)（公開日: 16/12/02）
- [OnTimeやメールDB、リソース予約DBで予約はあるのにノーツの空き時間検索で空いていると表示されるのは何故ですか?](#)（公開日: 16/10/21）
- [ユーザー個別の設定はどの設定画面で行いますか?](#)（公開日: 16/09/07）
- [Instant Client Refresherが動作して機能しているか確認する方法はありますか?](#)（公開日: 16/04/22）
- [OnTimeGC サーブレットが動作しているか確認する方法はありますか?](#)（公開日: 16/03/29）
- [OnTimeGCタスクが稼働し、メールDBと同期しているか状態を確認する方法はありますか?](#)（公開日: 16/03/25）
- [ブラウザで「Error getting Token: Not able to open Config Database」と表示されて画面が開かない。](#)（公開日: 16/03/24）
- [ノーツ内でOnTimeブラウザクライアントが起動しない場合の確認箇所は?](#)（公開日: 16/03/21）
- [サーバーとクライアントを流れるトラフィックデータについて](#)（公開日: 16/02/08）
- [クライアントがRequestを送信する際のコマンドやパラメータ及びデータサイズの調べ方](#)（公開日: 15/12/01）
- [OnTimeタスクの異常終了の検知方法について](#)（公開日: 15/10/06）
- [ユーザーの使用状況の確認方法](#)（公開日: 15/03/27）
- [OnTimeノーツクライアントが特定の人だけ起動しない場合のトラブルシュート。](#)（公開日: 14/05/26）