

を目指して

# コスト・性能・運用の課題を一気に解決！ AWSで実現したクラウドへの商用データベース移行 ～ MS SQL Server 2008R2からAmazon Auroraへのマイグレ～

2017年9月21日  
株式会社NTTスマイルエナジー  
技術システム開発部  
林田 悠基



# 自己紹介

---



林田 悠基 (ハヤシダ ユウキ)



株式会社NTTスマイルエナジー

技術システム開発部  
システムチーム リーダー

## 現在の業務 (2014年7月より)

- ・開発/運用チームのマネジメント
- ・社内外のシステム開発、運用全般
- ・新サービス、技術の調査

## (現在)好きなAWSサービス

- ・AWS Lambda
- ・AWS Database Migration Service



# 目次

---

1. NTTスマイルエナジーのご紹介
2. なぜDBシステム移行を決意したか
3. AWSへのDBマイグレーション  
(MS SQL Server 2008R2→Amazon Aurora)
4. 移行してみた感想
5. 今後の展望



---

# 1. NTTスマイルエナジーのご紹介



# NTTスマイルエナジーについて

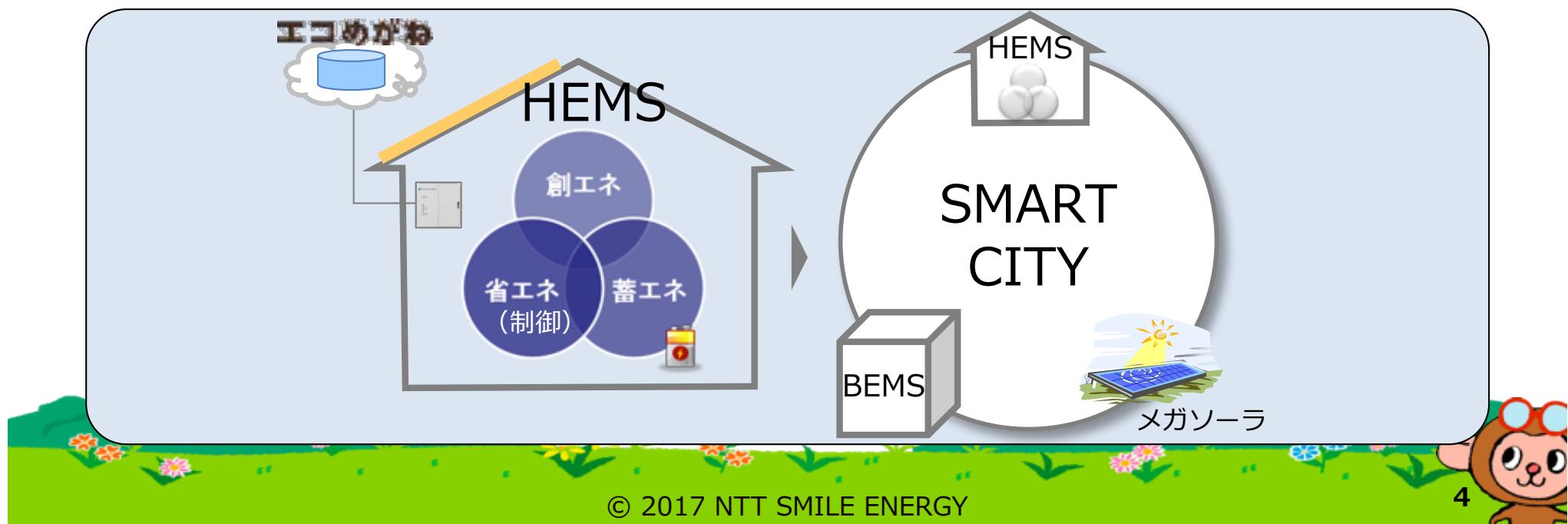
NTT西日本の通信技術とオムロンの制御技術により、エネルギー情報と通信を融合したサービスを提供し、来るべきスマートグリッド社会の実現に貢献するため、2011年6月にNTTスマイルエナジーを設立致しました。



通信ネットワーク技術  
クラウドサービスの提供  
BtoCサービスに関するノウハウ

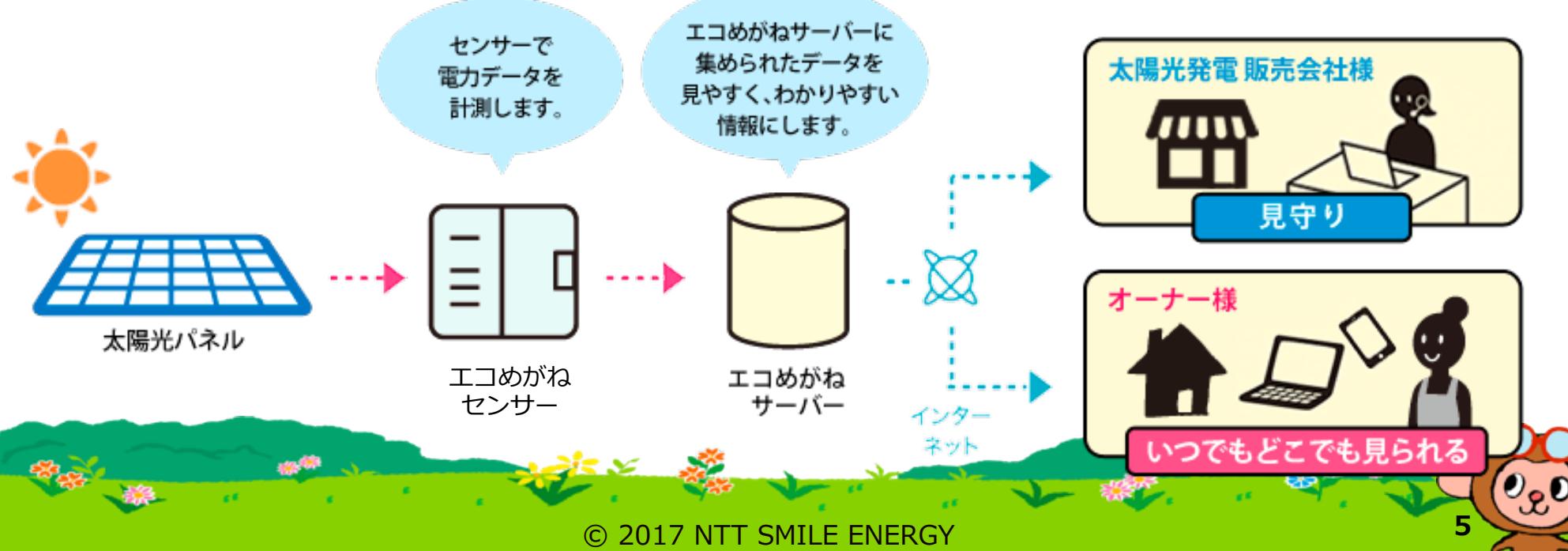


エネルギー見える化機器の製造  
省エネ手法に関するノウハウ  
パワーコンディショナーシェアNo1



# 情報×エネルギー = 「エコめがね」

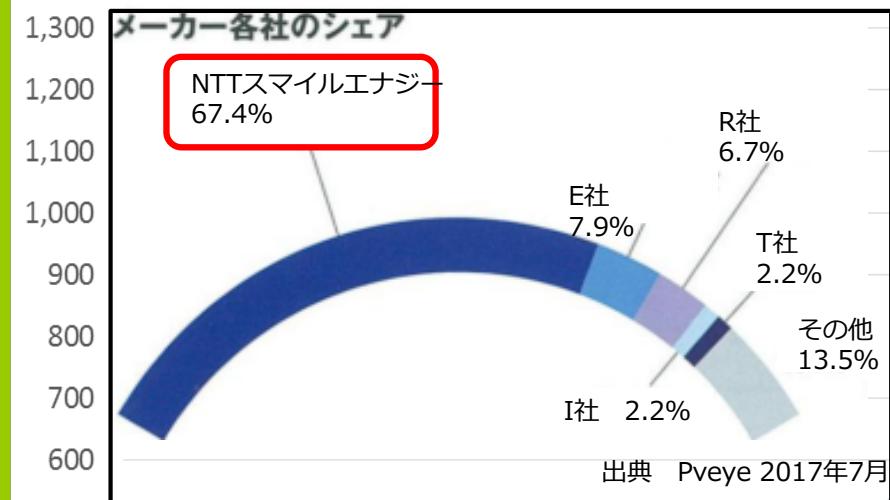
エネルギーをもっと身近にする3つのサービス



# 「エコめがね」の広がり

2017年6月現在、太陽光遠隔監視サービスである「エコめがね」は約4万力所、約1.2GW（原子力発電所一基分相当）の発電所に繋がっています。

## エコめがね見守り設備の推移



1.2 GW

1000M

700M

330M

83M

5M

2012年  
11月

2013年  
11月

2014年  
11月

2015年  
11月

2016年  
11月

2017年  
6月

## 過去の受賞歴

2013年12月



2014年10月



2016年5月



---

## 2. なぜDBシステム移行を 決意したか

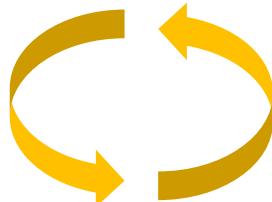


# 度重なる障害の発生

利用者数が増加していくにつれ・・・



Webサイトの表示が遅い！  
データの計算が間に合わない！  
センサーデータが受信(保存)できない！



緊急対応

SQLクエリ (処理ロジック) の見直し  
DBサーバのスペック変更 (SATA→SSD)  
INDEX貼り直しの実施  
統計情報の定期的な更新実施 ...etc



直しては次の新たな問題発覚・・・を繰り返す日々



# 開発工数の増加

---

場当たり的なアプリ改修を繰り返したことで・・・



改修によるDB処理影響の把握が困難に！  
= 小さな改修でも全疎通試験が必須

テスト工数が一向に減らない！  
(むしろ増えていくデススパイラル・・・)



# DBシステムが抱えていたリスク

---

トライアルから正式サービスへ突入したので  
機能リリースで手一杯状態のため、  
可用性・拡張性などを全く考慮できていなかった。。。。

## DB最適化を考慮していないアプリケーション

- ・適切なINDEXが張られていないテーブルへの全scan
- ・パーティション分割されない巨大テーブルへの常時scan
- ・高頻度でのDELETE/INSERT繰り返し処理

## 1台のマスタしかないDBシステム

- ・復旧手段はバックアップ＆リストアのみ
- ・リストアも手順が不明確で復旧できるか未確認状態！



**DBシステムがサービス停止の最大リスクに！**



---

こうなつたら  
DBシステムを変えるしかない！

と思い立ったものの・・・



DBの運用が  
変わって大変にな  
らないかな？

DBのコストが  
逆に増えないかな？

性能的に他のDBで  
大丈夫かな？



安くて、高性能、楽チン運用  
そんなDBないかなあ～？  
(ワガママな悩み)

そして出会ったのが  
**Amazon RDS**



---

ただし！  
いきなりDB切り替えはアプリケーションの改修が  
必要なので相当敷居が高い！

そこで・・・

リフト&シフト作戦！

今のDBをRDSに常時同期して  
今後開発するサービスはそちらをマスタに！  
いざれば新DBを定義含めて再構築し、  
新DBに最適化した形でデータ移行しよう！



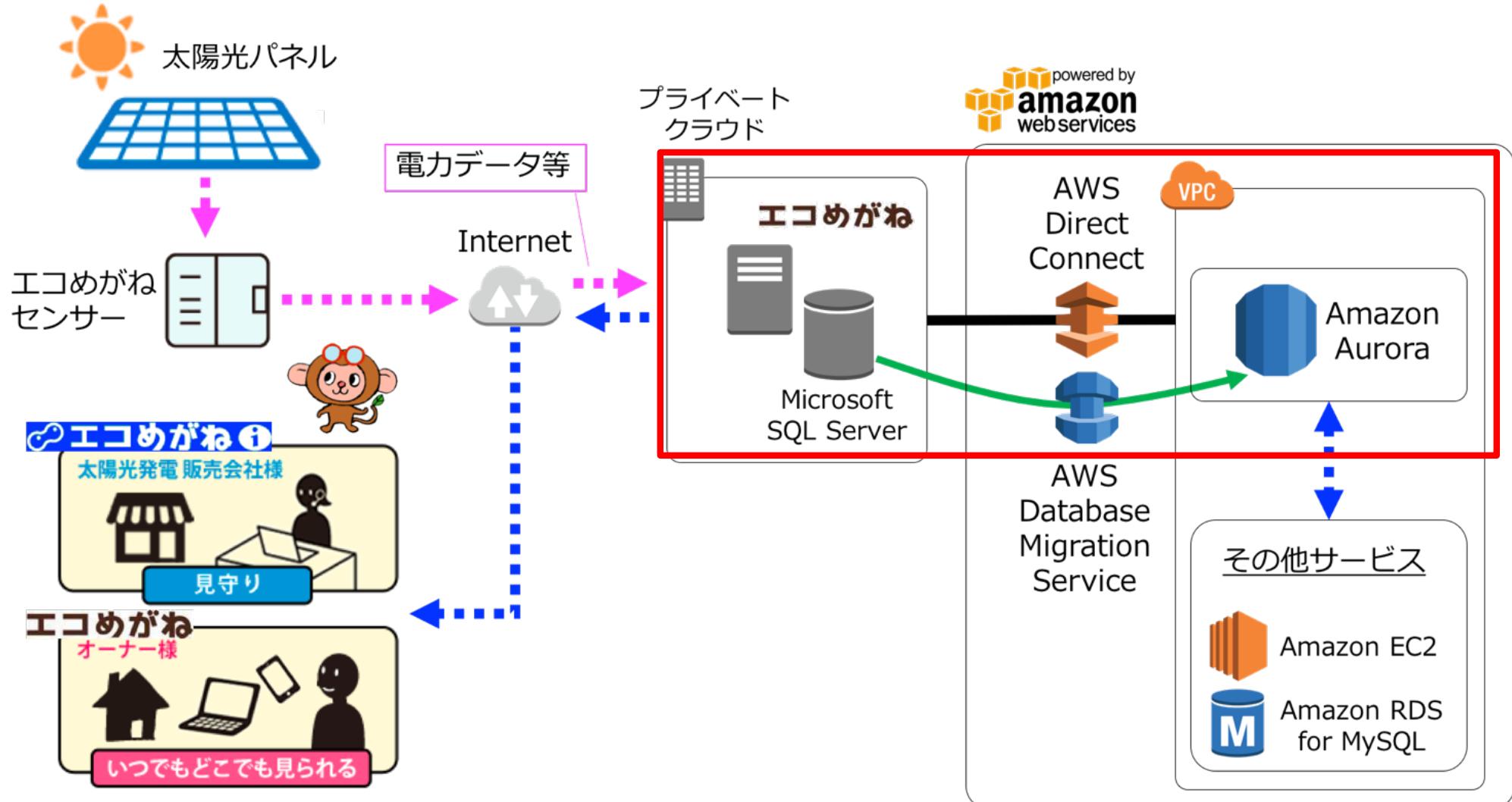
**AWS Database Migration Service  
で実現可能！** (と聞いた)



---

### 3. AWSへのDBマイグレーション (MS SQL Server 2008R2→Amazon Aurora)

# システムの概要



# そもそもなぜ移行先がAurora ?

- DBミドルウェアの**ライセンスコストがフリー** !  
→将来はわかりませんが・・・
- AWSが**積極的に開発をしているサービス**なので  
**今後の拡張や性能向上が非常に期待**できる !  
→頻繁なアップデートについていくのは大変ですが・・・
- デフォルトで**高い可用性 & 冗長構成も容易に実現** !  
→シングル構成はもう嫌だ・・・

etc...

何よりも  
クラウドを使うなら「トレンド」を使いたい !  
~~(という邪な思い)~~

もちろん移行は大変でした orz



# Aurora移行で直面した課題と対策



大小含め、様々な技術課題が発生。  
結局、移行を決意してから移行完了まで半年かかりました...

(課題の一例)

- SQL Server側の設定不足による同期実行エラー
- トランザクションログの取り扱い方式の違いによるディスク容量圧迫
- SQL構文の不適切な処理実施による同期停止エラー
- ネットワーク回線品質の問題によるスループット低下
- Auroraインスタンスの性能ボトルネックによるスループット低下

**AWSサポート、AWSパートナーさんのおかげで  
全ての課題を解決できました！**



---

## 4. 移行してみた感想



- 既存アプリやDBそのものの制約は出たが、意外となんとかなった印象

→弊社の利用状況・環境においては、です。



- 性能に関してはまだ正直わからない

→これから詳細な検証を行う予定だが、テーブルスキャンが速くなった印象

- AWSはDB移行を支援するツールやサポートが充実してる

→コストも期間も抑えつつ移行できた！  
5年ぐらい運用してたDBをたった半年で別DBに移行できた！

- 念願の冗長化をクリックひとつで実現できたのはすごい！

→Auroraのディスク冗長化 + マルチAZ設定でかなり高い可用性を持ったDBに生まれ変わった！



---

## 5. 今後の展望



# 今後の展望

- 既存システム基盤のAWS移行

※但し、当面はプライベートクラウドとの併用を想定

→アプリを「AWSネイティブ」な機能へ更改

活用を検討しているAWSサービス



Amazon  
ECS



AWS  
Lambda



Amazon  
Kinesis  
Firehose



Amazon  
Athena



Amazon  
API  
Gateway

- ビッグデータの更なる活用

→データ分析基盤の構築

活用を検討しているAWSサービス



Amazon  
Redshift



# 今後DB移行を検討される皆様へ

## ① DBインスタンスはケチらず大きめのものを選択しましょう

→その方がレプリケーションも早く終わります！

OnGoing（常時同期モード）中でもインスタンスは  
変更できるので、後でスペックダウンできます！

※途中変更出来ないと勘違いしてケチった結果、長期化・・・

## ② 回線はDirect Connect等の専用線サービス利用を推奨します

→ケチってインターネット回線を使うと

レプリケーションがうまくいかないことがあります

※DB同期にインターネット回線は使うべきでは無い！（と思います）

## ③ 諦めてアプリ(SQL)改修でカバーすることも覚悟しましょう

→アプリケーション改修は覚悟の上、移行しましょう。

## ④ AWSサポートおよびAWSパートナーを頼りましょう

→幅広いDBの知識でサポートしてくれるので安心です。

※アプリは自社でなんとかする必要があります



