

なぜ私たちは今、ダウンタイム・ゼロにこだわるのか

—業務現場、自動制御、Connected Device、情報系/制御系連携、基盤系...

今、そしてこれからのシステム像から信頼性の条件と可能性を考えます

2011年9月8日

札幌スパークル株式会社

桑原 里恵

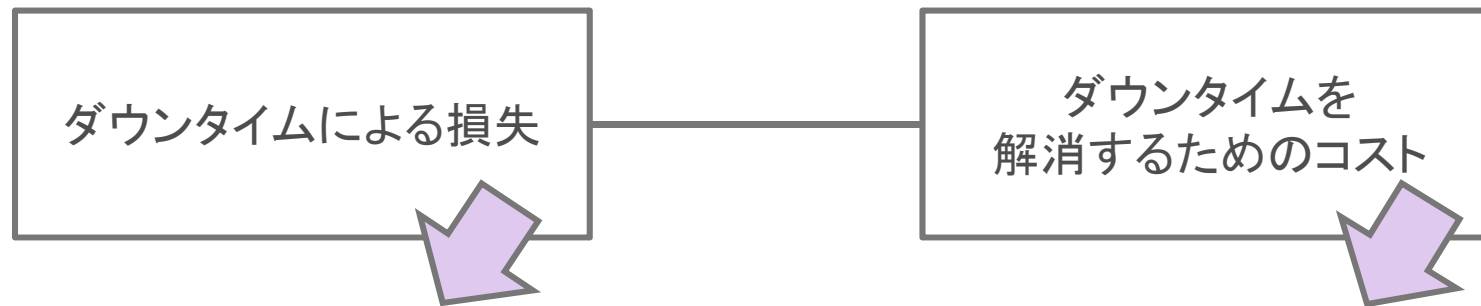
Twitter @SatoeKuwahara

マラソンテクノロジーズ
「ダウンタイムゼロを考える」セミナー

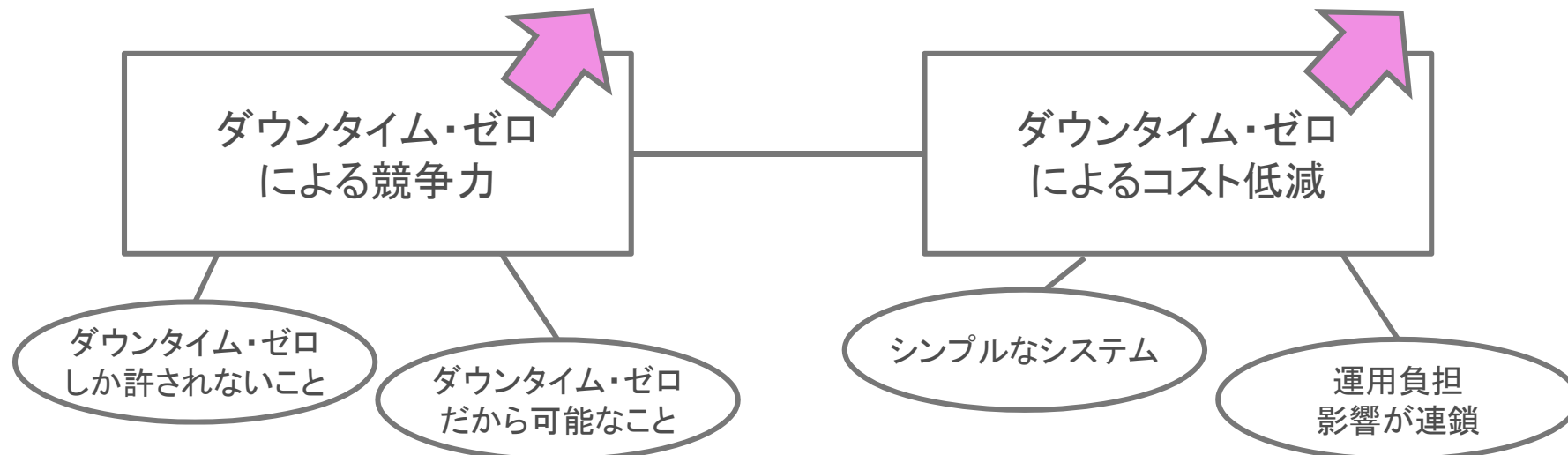


「信頼性とコスト」のトレードオフを超えて

ずっと考えてきたトレードオフ・・・“守り”のアプローチ

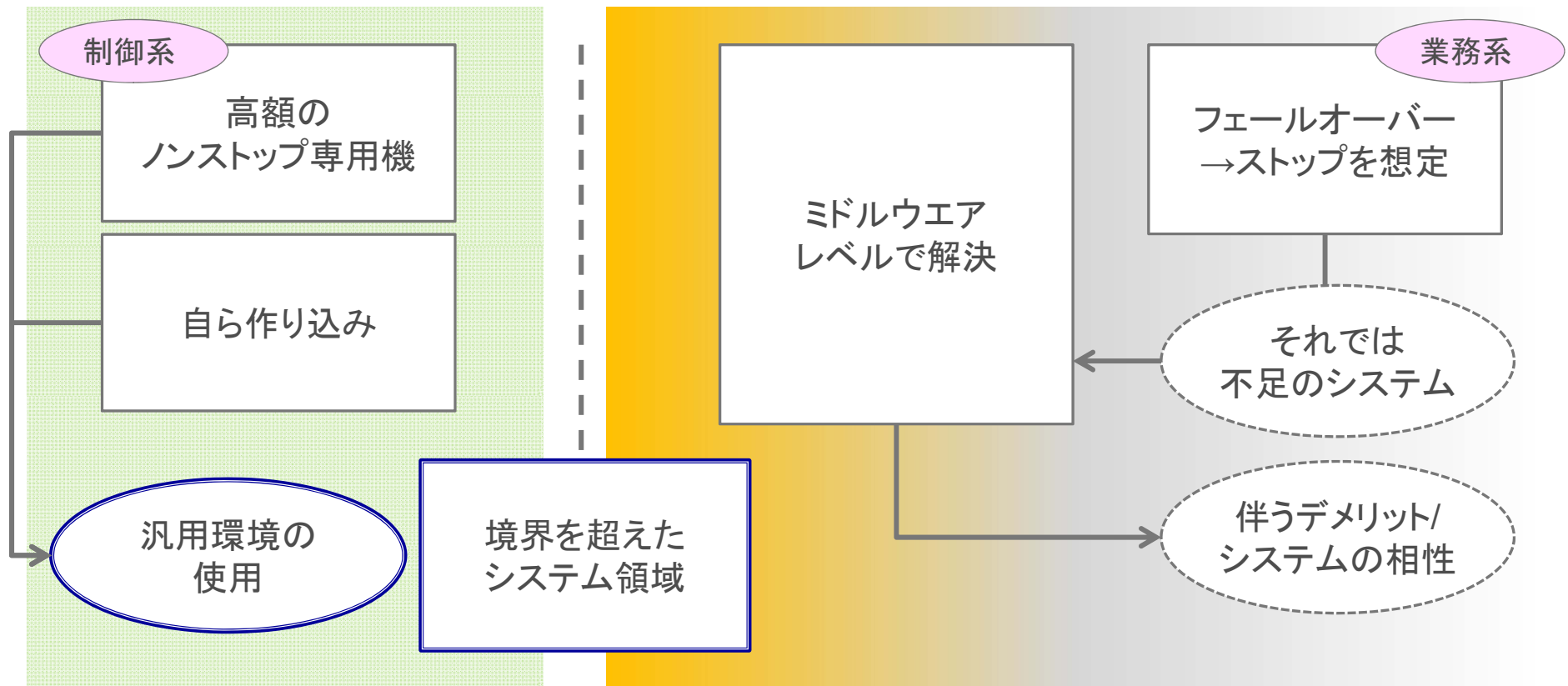


今、トレードオフが変わってきた・・・“攻め”へ



HAは「システム特性で使い分け」してきた。しかし...

『High Availability』のマイナス側面 → 使い分けで乗り越える
2つの視点； ①システム特性 ②費用とのバランス



今、「汎用環境の使用」「境界を越えるシステム」、そして連携へ

なぜダウンタイム・ゼロを求めるのか？

「High Availability」なのか、「Zero Downtime」なのか・・・

なぜ、ダウンタイム・ゼロを求めるのだろうか？

どんなシステムで求められるのだろうか？

1. 処理の中断が致命的である(品質、時間を直撃)
2. 「データの抜け落ち」、その1行が致命的である
3. 処理及び稼働状態の復旧が困難、又は時間がかかる
4. データの復旧が困難、又は時間がかかる
5. システム間が連携、影響先が多大、連携先が相互

従来； システム単位の処理とデータの問題

そして今； 連携するデバイス、マシンとデータの問題

さらにこれから； 連携するシステムと全体システムの問題へ

限られた、しかし多くのシステムに必要。そしてさらに・・・

■ ダウンタイム・ゼロを必要とするシステム

限られた対象、しかし多様な分野で多くのシステム
しかも今、『組合せ』（連携、統合）で急拡大

- － 最前線の現場システム、自動制御
- － 監視系、基盤系 * 盲点; 監視系が弱点で止まるシステム多数
- － 情報系/制御系連携、企業間連携、リアルタイム・プロセス
- － Connected Device、クラウド連携/Web2.0、社会基盤

■ 今、直面しているもうひとつの問題

多重に重ねたHA機構・・・ミドルウェア層、HW層

- － 異なる特性のHA→相互に干渉、無駄な動作、整合設計が必要
- － システム全体が複雑化→不利益; 障害、運用、保守、パフォーマンス
- － バージョン間の相性、想定使用年数の食い違い

システムの可能性とTCOの観点→シンプルであること

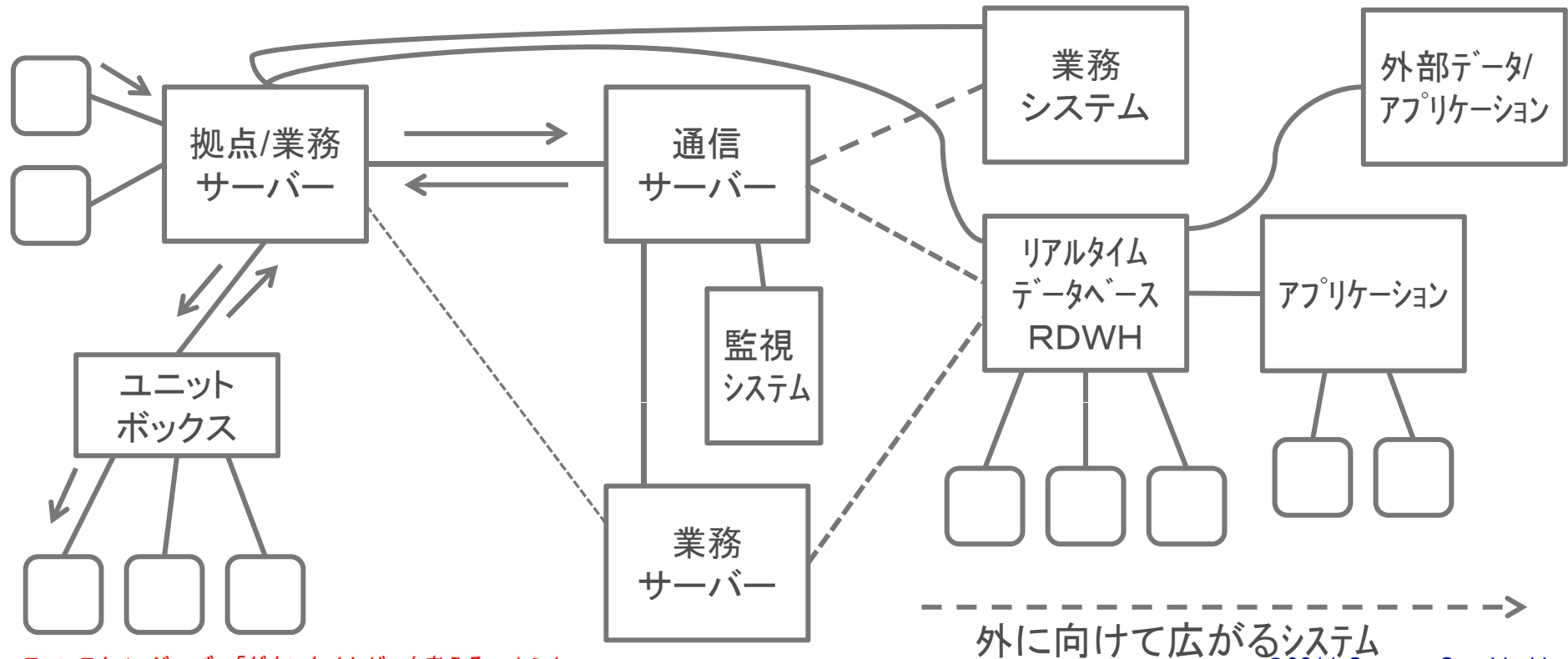
→信頼性の確保が、**複雑性**や**システム使用制約**になってはいけない

そして、連鎖するシステム・・・リアルタイム・プロセスへ

■ 自律したシステムが組み合わさって、ひとつのプロセスを成す

- 共通認識； 個々の自律性を確保。個々が信頼性を担保する(もたれ合わない)
- 共通認識； 提供側が信頼性を担保する→プル側に「万一の処理」「チェック」を前提にせず
- 共通認識； 信頼性は全体のバランス。「万一」には“頭戻し”

★すべては複数が連携することを前提に、総和をシンプルにする思想



まとめーデザインの視点でダウンタイム・ゼロを考える

- ダウンタイム・ゼロを求めるシステムが増えてくる
 1. ダウンタイム・ゼロを競争力につなぐ→自ら“前提条件化”
 2. より大きなシステムの最前線に→多数のデバイス、DB連携
 3. IT的な基盤化、システム相互の連携へ→制御系/情報系の統合へ
 4. シンプルであることを強みに→多重層、無駄なヘッジを解消
 5. クラウドや社会基盤との連携へ拡大 例;“スマート**”の構図

業務、システムの両面から

“約束した信頼性を確実に達成”することが大前提に
 同時に、信頼性は全体のバランスで決まる
 “技術の力”で“信頼性のバランス”をブレイクスルーする

●デザインの視点から“ダウンタイム”の問題を考えてみよう

- ー 全体像とシステム間の関係性から求められる信頼性
- ー 逆に、信頼性を確約することで生まれる可能性
- ー そして、シンプルであることによって可能になる世界