

ウ) ネットワークサービス制御技術 トピックと研究進捗

NEC

2005年11月28日

研究トピック

「1-4.大容量アドレス検索・変換技術」

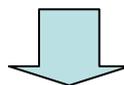
スケーラブルコンテキストプラットフォーム
の研究開発

背景

- 近年、実世界の情報を電子的に取り込み、それらの情報をさまざまなサービスへ応用することを目的として、RFID (Radio Frequency Identification)、センサが注目を集めている。
 - 現在、物流管理、物品管理、食品トレーサビリティ、医療、アミューズメントなどさまざまな分野での応用が期待されており、すでに各業界で実証実験が始まっている。
- RFID、センサの普及に伴い、RFIDタグ、RFIDタグリーダ、センサの種類が多様化、RFIDタグリーダが読み取るイベント、およびセンサからのセンシング情報の増加、そしてそれらの情報を利用するアプリケーションの増加などが考えられる。



- これら大量の情報の中からアプリケーションに有用な情報を検出、加工し、アプリケーションに通知するプラットフォームが必要となる。



- 我々はこれらを実現するプラットフォームをコンテキストプラットフォームと呼び、その方式を実現する。

センサーイベント、コンテキスト情報の爆発

- 2005年のウォールマートの実証実験
 - 一部の小売店・配送センターにおいて、一部のパレット／ケースにだけRFIDが付けられたにもかかわらず、年間11 millionのRFIDイベントが発生

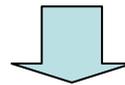


- 本格運用したらRFIDイベントはさらに膨大な量に

センサーイベント、コンテキスト情報の爆発

- Ubilaにおける想定

- ユビキタス社会においては100億端末がネットワークに接続

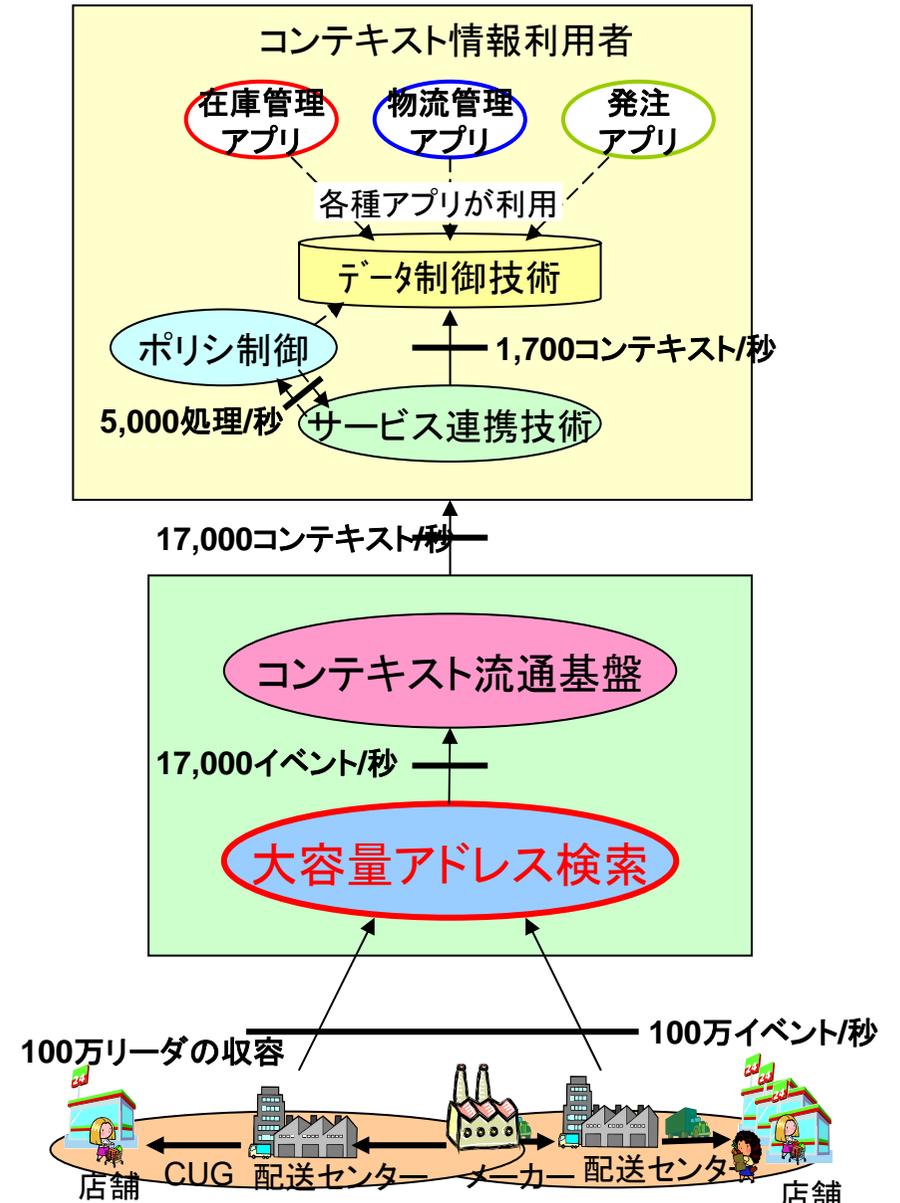


- 実証実験レベルのユビキタスサービスではイベントの量は限られており、サーバー台で十分に処理できる。
- しかし、巨大インフラによる本格的なユビキタス環境では、あらゆる場所にセンサーが配置され、そこから発生するイベントは膨大な量に

ウ) 内での本研究テーマの位置づけと想定ユースケース

[RFIDトレースシステム]

- 100億個のRFIDタグを付けられた商品を管理する物流管理を想定
- ある地域内 (店舗、倉庫、工場、トラックなど) にある商品群の状態をリアルタイムに管理する。
- RFIDリーダは、店舗の棚ごと、倉庫のラックごと、トラックごとなど 100万台を想定。
- RFIDには、製品名や製造日、製造工場など 100属性程度と対応していると想定。
- RFIDリーダは、1秒ごとRFIDタグを読み取り、サーバはAPIに応じて必要な統計情報を返し、データベースに記録
- データベースに記録されたデータは各種アプリから利用すると想定
- ネットワーク上には複数のコンテキスト情報利用者や複数のコンテキスト業者の端末やサーバが総計 100億台接続されていると想定



関連技術：複合イベント処理(CEP)

- 人間の活動(高位レベルイベント)はデバイスから発生する低位レベルのイベントから推測することができる
 - 低レベルイベント:RFID読み出しイベント、センサー情報
 - 高レベルイベント:物流トレーサビリティ、輸送中の温度管理違反
- 低位レベルのイベントから高位レベルのイベントを検出する技術体系が必要。



- 複合イベント処理(CEP: Complex Event Processing)[1][2]
 - 複数のイベントに基づいて、コンテキストの処理、加工、通知を行う技術
 - イベントトラヒックの中から重要な低レベルイベントを検出し、それらの意味を要約する一つのイベントを生成することを**イベントアグリゲーション**と呼ぶ。
 - 共通のデータやタイミングによって**複数のイベントからなるパターンを表現する能力**
 - **高機能なルール記述**



- **膨大な量の低レベルイベントトラヒック**の中から、リアルタイムでイベントのパターンを認識する技術が課題

[1] D. Luckham, “The Power of Events”

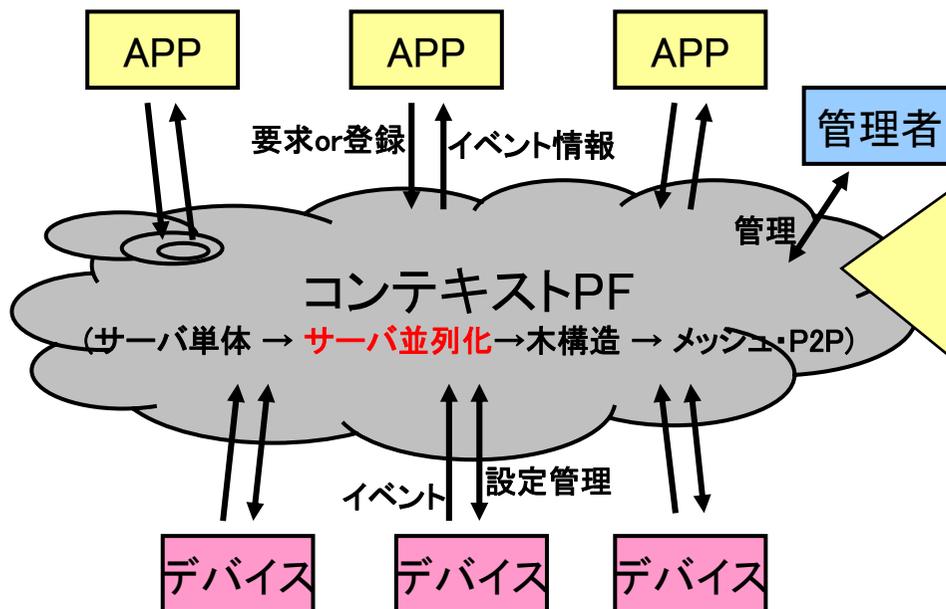
[2] M. Mansouri-Samani, “GEM: a generalized event monitoring language for distributed systems”

コンテキストプラットフォームの要求条件

- 開発するコンテキストプラットフォームは以下の要求条件を満たすことを目的とする。
 - 1. スケーラビリティ
 - プラットフォームはデバイスからの**イベントの増加**、**アプリケーションから設定されるルールの増加**などに対して、安定したサービスを提供する。すなわちプラットフォームは、通知すべきイベントを漏れなく、**アプリケーションの要求する時間内に通知**する。
 - 2. 複合イベント処理
 - 単純なIDフィルタリングだけではなく、**複合イベント処理により、イベントのより高度なフィルタリング/処理/通知**を行う。
 - 3. プッシュ型サービス
 - アプリケーションはあらかじめ通知条件（以降、ルールと呼ぶ）をプラットフォームに登録しておき、ルールにマッチするイベントをアプリケーションに通知するサービスを想定する。

コンテキストプラットフォームコンセプト

- RFIDリーダ、センサから発生する膨大なイベントを収集、取捨選択、加工、蓄積し、アプリケーションの必要とするイベントのみをアプリケーションに通知する。



取組み内容・課題

- **フィルタ階層化/再構成/トラヒック集約**
 例)フィルタ1: 全店舗の特定リーダ (納品ゲートなど)
 フィルタ2: ある店舗の全リーダ
 → 用途に応じたフィルタ構成の実現
- **スケーラブルな負荷分散方式**
 - Dispatcherによる負荷分散 (Webの負荷分散などと異なり コンテキストの複合処理を考慮する必要)
- **大量のデバイスの機器構成管理**
 - ユビキタス時代には100万オーダーの機器管理が必要

プラットフォームサービスの分類

- リアルタイム系(プッシュ型)

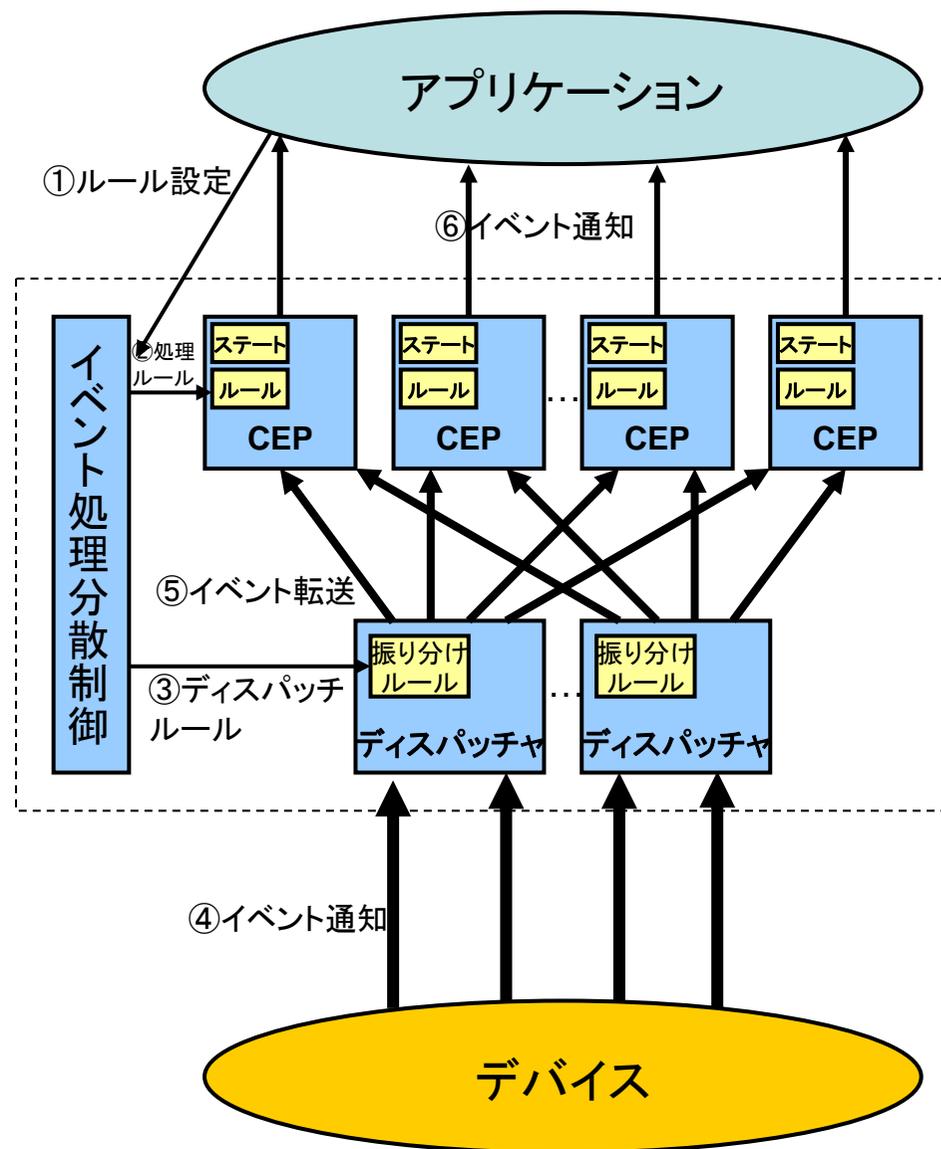
- イベントが発生すると、APIを通じ、即座にアプリケーションに通知する。
- 必要なイベント情報のみを取捨選択して通知する必要

- 蓄積系(プル型)

- イベントはDBにログとして蓄積され、アプリケーションが必要とした時に索引を行う。
- 大規模分散DBが必要

ディスパッチャ方式コンテキストPFの提案

- ディスパッチャにより複数のイベント処理サーバにイベントを振り分け
 - 複合イベント処理を行うため、関連するイベントは同一サーバに振り分けなければならない → **ルール分配アルゴリズム**
 - イベント処理制御部が、処理サーバに複合イベント処理ルールを設定するとともに、連携した振り分けルールをディスパッチャに設定
- イベント処理ルールに関連しないイベントはディスパッチャでフィルタアウト
 - 処理サーバの負荷軽減にもつながる
- ディスパッチャ自体の並列化も可能



ルール記述

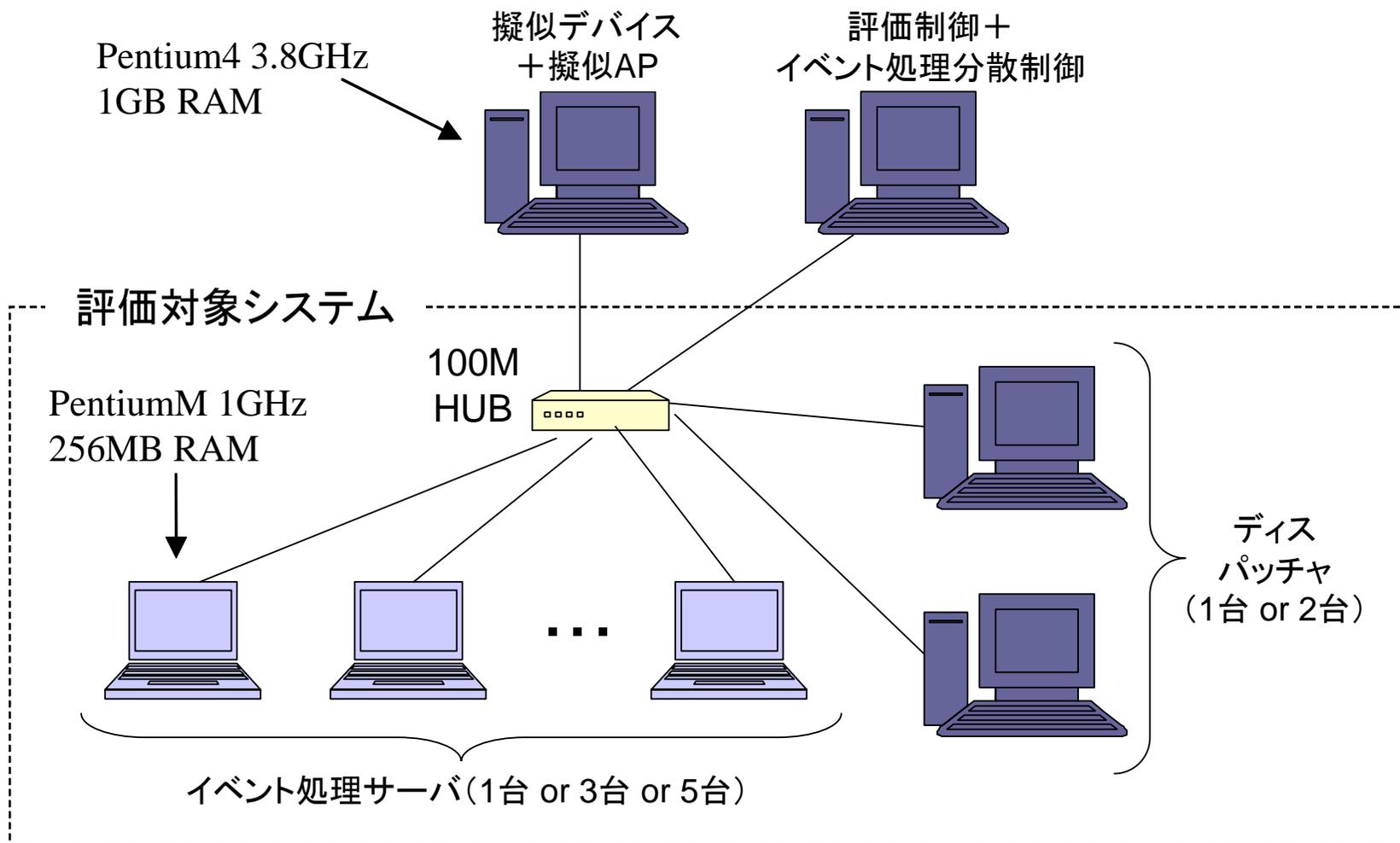
1. Rule (*Event A*, *Action X*)
– *Event A* が起きたら *Action X* を実行
2. Rule (*Event A* | *Event B*, *Action X*)
– *Event A* または *Event B* が起きたら *Action X* を実行
3. Rule (*Event A* & *Event B* within *t*, *Action X*)
– *Event A* と *Event B* が時間 *t* 内に起きたら *Action X* を実行
4. Rule (*Event A* → *Event B* within *t*, *Action X*)
– *Event A* の発生後時間 *t* 内に *Event B* が起きたら *Action X* を実行

- 高速化のためシンプルな複合イベント処理ルールを選択
– より高度な複合イベント処理は上位で行う
- Event項のネストも可能
– 例: Rule ((Event A | Event B) & Event C, within t , ActionX)

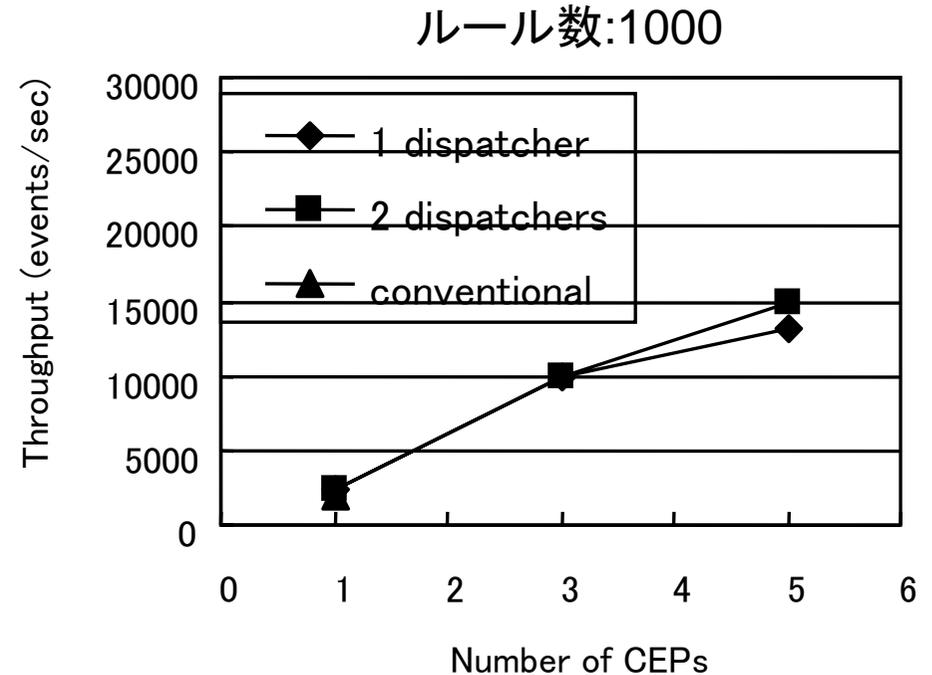
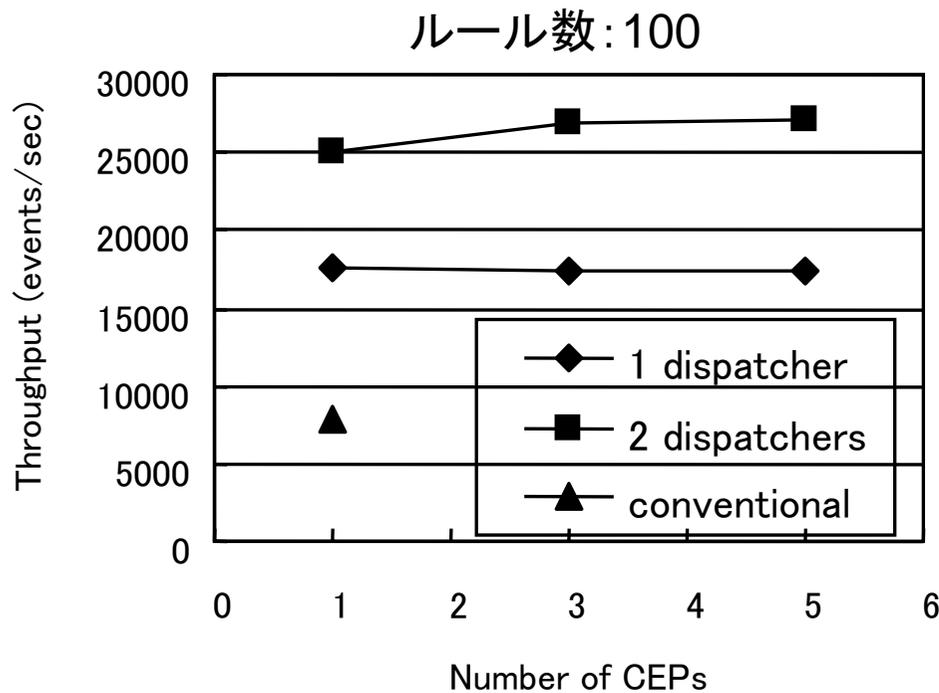
ルール分配アルゴリズム概要

- 各処理サーバの負荷が均一になるようにルール分配
 - 分配ルール数を均一にするだけでなく、ルールに関連するイベントの発生量も考慮
- 同一イベントに関する処理ルールは出来る限り同一処理サーバに分配
 - ディスパッチャにおけるイベントコピーを最小限に抑える

コンテキストプラットフォーム性能評価システム構成



性能評価結果(スループット)



- ディスパッチャ2台、処理サーバ3台構成時に25000イベント/秒以上を達成
- イベント処理ルール数は CEPでほぼ同数配置
- サーバ1台の時も基本構成より高スループット
 - ディスパッチャによるイベントフィルタリングにより処理サーバの処理量が軽減
- サーバ台数にスループットがスケール
 - 単体時2300イベント/秒に対し、サーバ5台時15000イベント/秒
 - 構成台数を増やすことにより更にスループット向上が可能

性能評価結果(遅延時間)

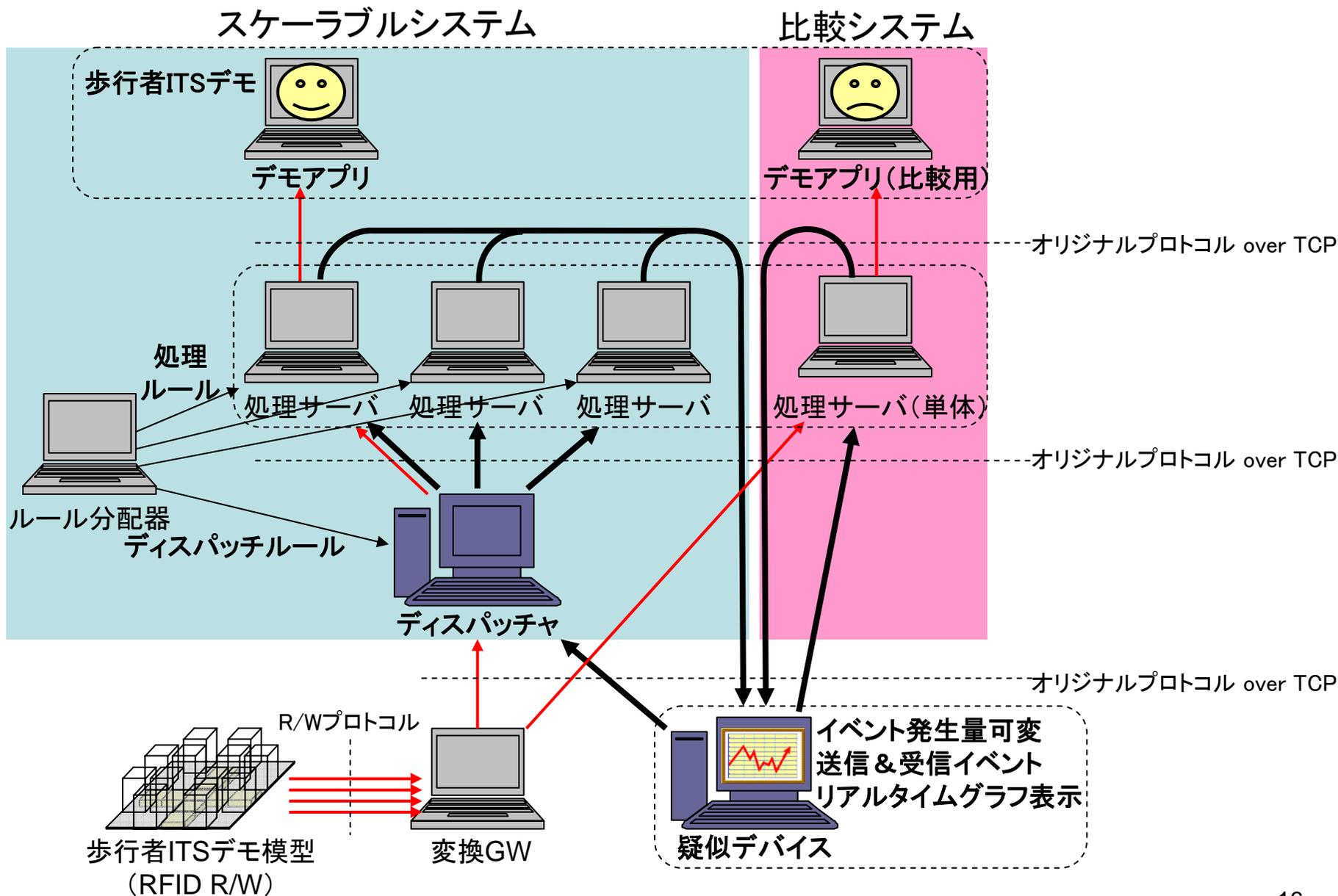
非発散	発散
-----	----

ルール数100のときの遅延時間

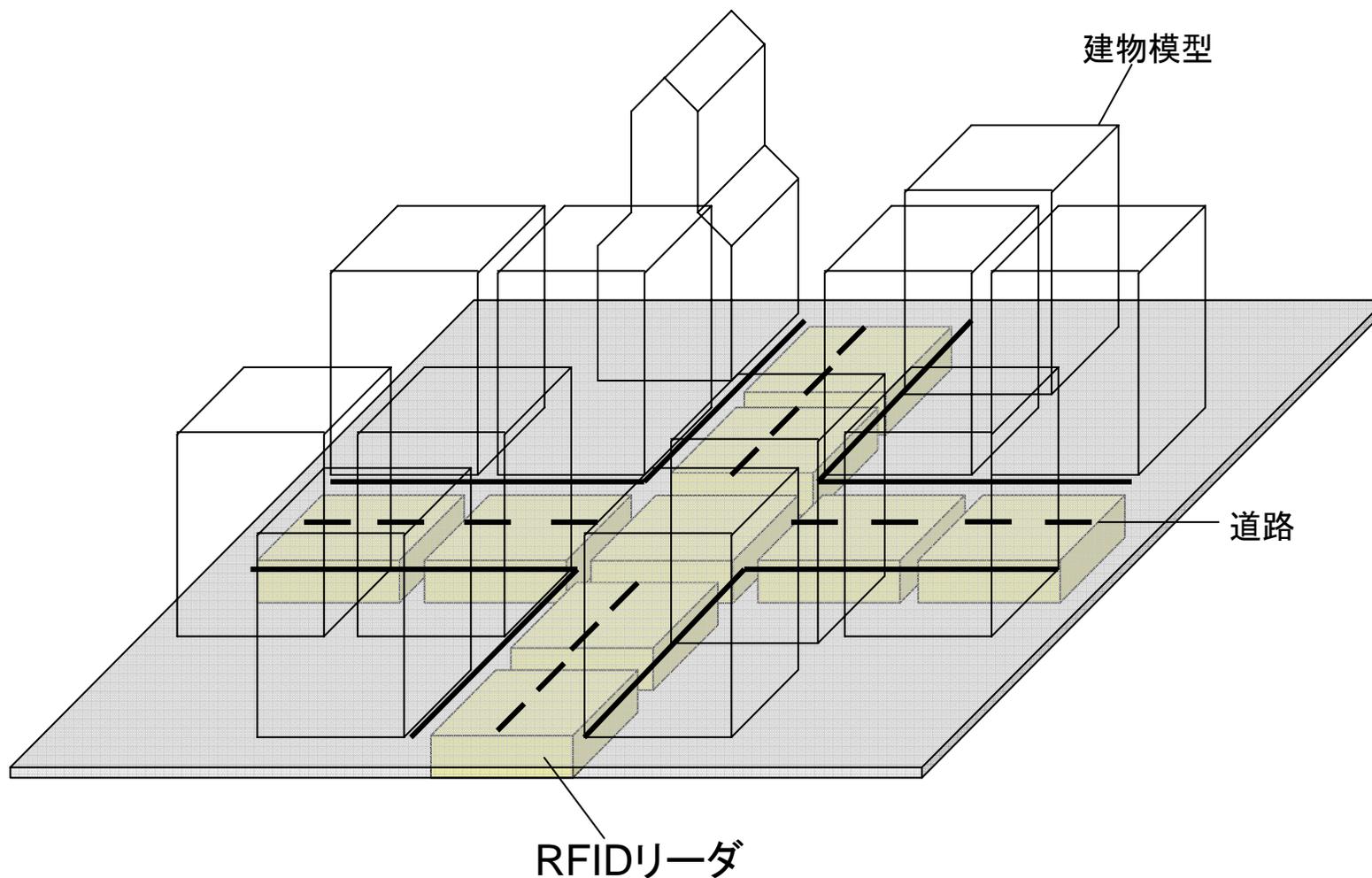
		event generation rate (events/sec)						
		5000	10000	15000	20000	25000	30000	35000
system configuration	1 EP + 1 dispatcher	0.132811	0.129483	0.124879	15.00042	43.08938	67.53679	89.99679
	3 EPs + 1 dispatcher	0.131144	0.130185	0.130822	15.97324	44.61663	67.11239	85.60402
	5 EPs + 1 dispatcher	0.140947	0.127496	0.128533	15.21658	45.36814	68.70222	84.23073
	1 EP + 2 dispatchers	0.164971	0.145098	0.139267	0.132105	0.131579	32.14656	74.03827
	3 EPs + 2 dispatchers	0.167703	0.166867	0.157857	0.149594	0.145469	15.15305	34.9421
	5 EPs + 2 dispatchers	0.146575	0.172262	0.166233	0.163261	0.158341	14.48551	35.9918
	conventional	0.102945	87.15411	290.1045	428.4373	556.4184	650.6411	772.5818

- イベント処理サーバ数を増やすことにより、基本構成の場合と比べて、発散 (i.e. CEP待ち行列溢れ) しないイベント処理量の区間が広がる
- 発散しない場合の遅延時間は0.103秒から0.172秒となっており、トレーサビリティなど想定するアプリケーションの要求を満たすと考えられる
- イベント処理量が増えても、処理サーバの台数をさらに増やすことにより遅延を抑えることができる

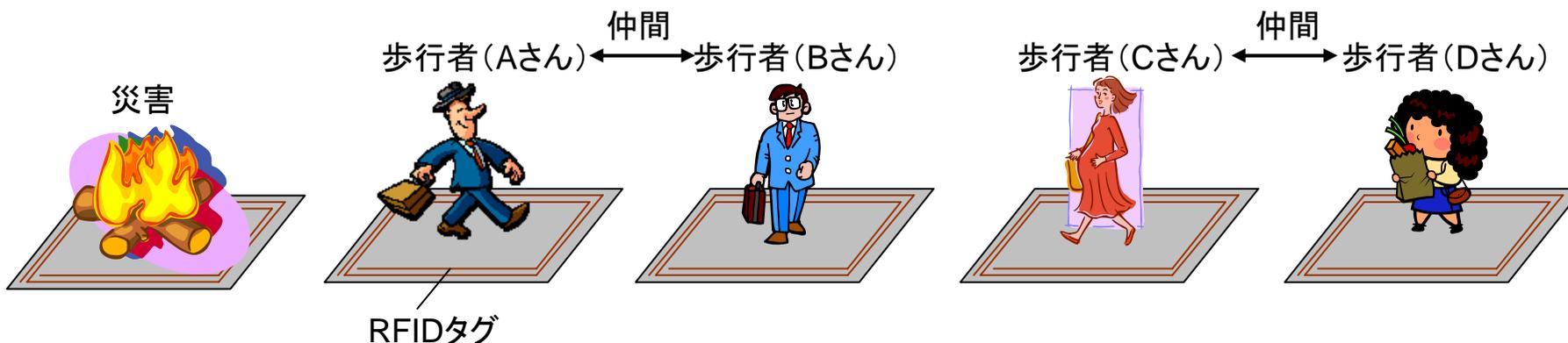
スケーラビリティデモ開発



歩行者ITSデモシステム(インフラ模型)



歩行者ITSデモシステム(歩行者模型 & 画面)

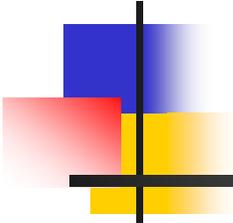


- 歩行者が移動すると、過去の位置と現在の位置を比較して方向を通知
- 仲間が1ブロック以内にいると通知
- 歩行者が災害に近づくとアラーム



まとめ

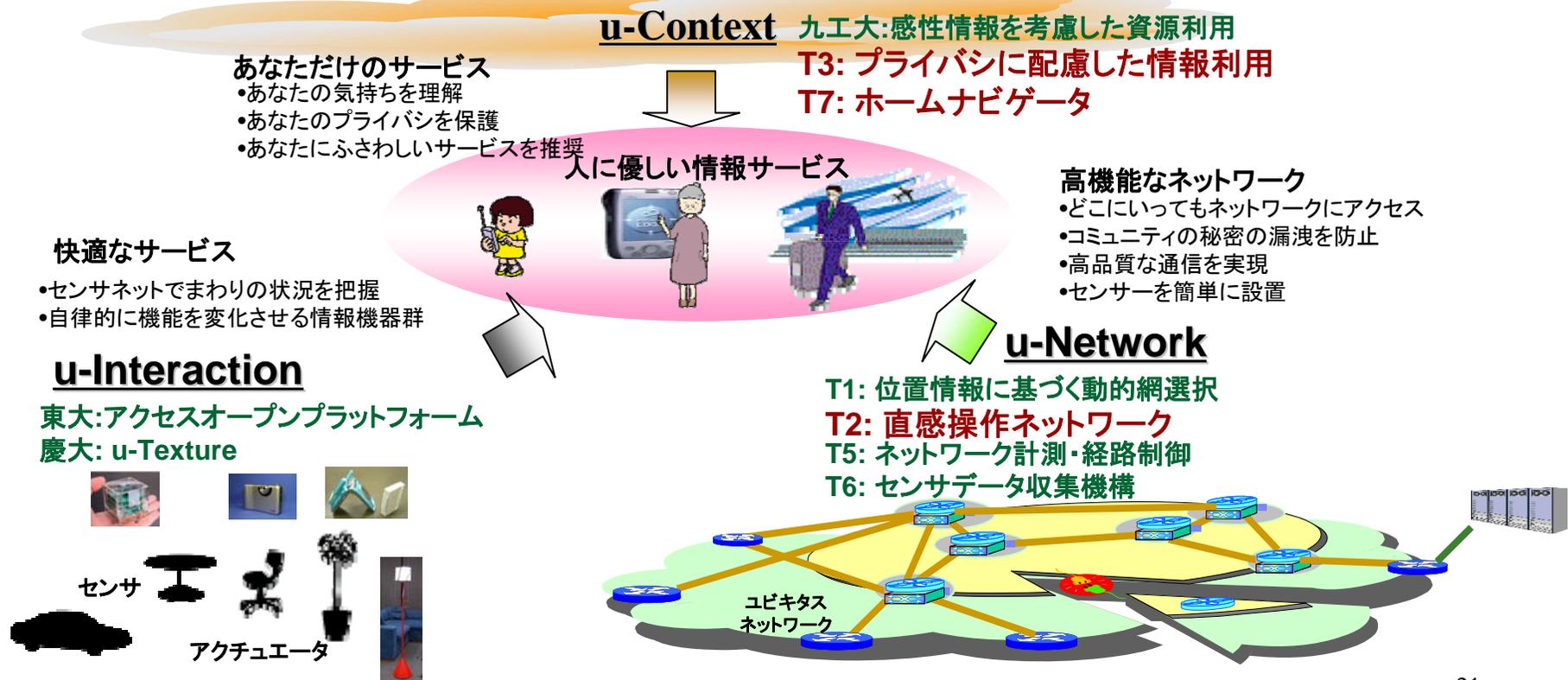
- RFID、センサの普及に伴い、それらから来る大量の情報の中からアプリケーションに有用な情報を検出、加工し、アプリケーションに通知するプラットフォームが必要となる。
- ディスパッチャ方式のコンテキストプラットフォームを試作
- 性能評価を行った。
 - ディスパッチャ2台、処理サーバ3台構成時に25000イベント/秒以上を達成
 - サーバ台数にスループットがスケール
 - 発散しない場合の遅延時間は0.103秒から0.172秒となっており、トレーサビリティなど想定するアプリケーションの要求を満たすと考えられる
- 性能比較用歩行者ITSデモを開発
 - 高負荷時の提案方式の優位性を確認



研究進捗

他組織との連携実証デモ及び
NEC 委託研究

- 「十人十色を実現する“小粋な”ネットワーク・サービス」を目指します。
 - **u-Context**: 状況に合わせて、あなただけのサービスを実現
 - **u-Network**: ユビキタス時代を支える高機能なネットワークを提供
 - **u-Interaction**: あなたの状況を理解し、サービスを提供する知的空間を構築



T2: 直感操作ネットワーク

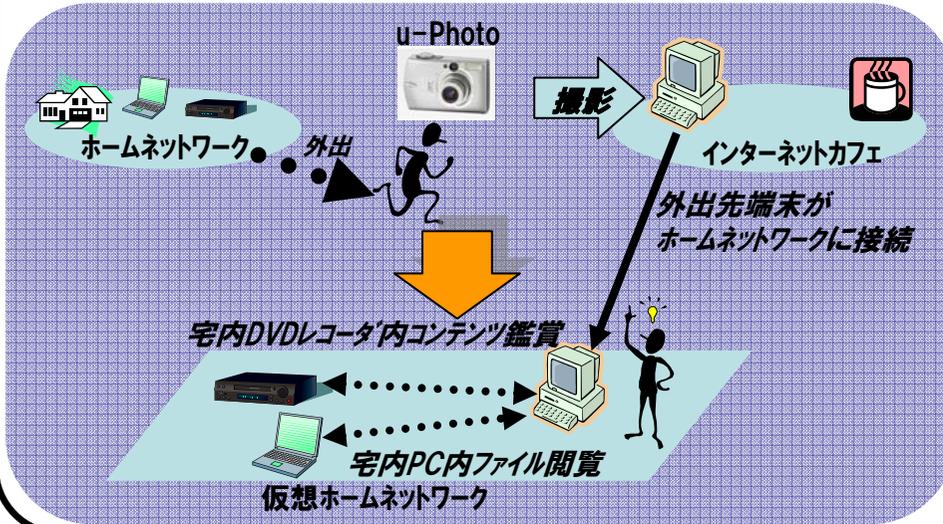
カメラをUIとする直感的操作によりネットワーク環境の容易なカスタマイズを実現

1. 仮想ネットワークに接続させたい機器をカメラで撮影
 2. 接続させたい仮想ネットワークを選択 → 仮想ネットワークへの接続完了
- リモートアクセスや仮想網構築に従来必要だった複雑設定を一切排除

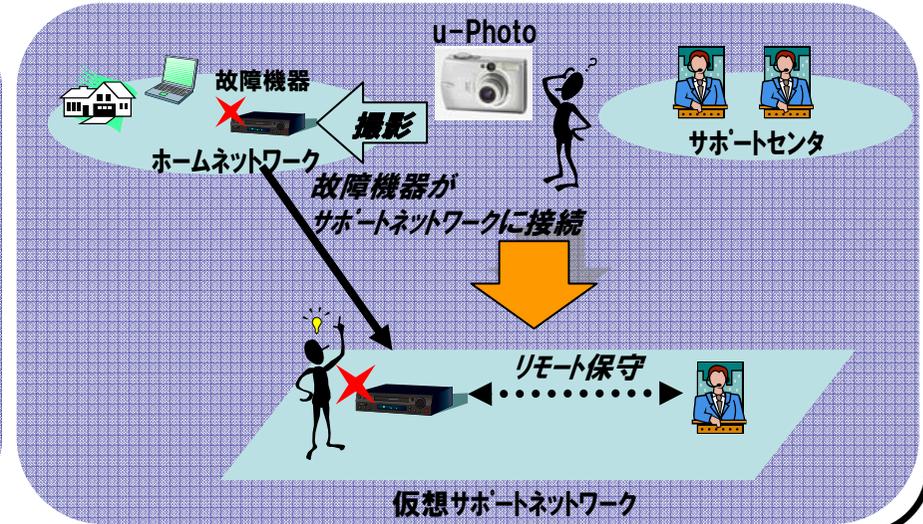
NECは仮想ネットワーク構築技術を提供

ユースケース

簡易ホームネットワーク拡張



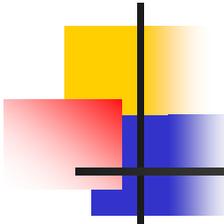
簡易リモート保守



システム構成



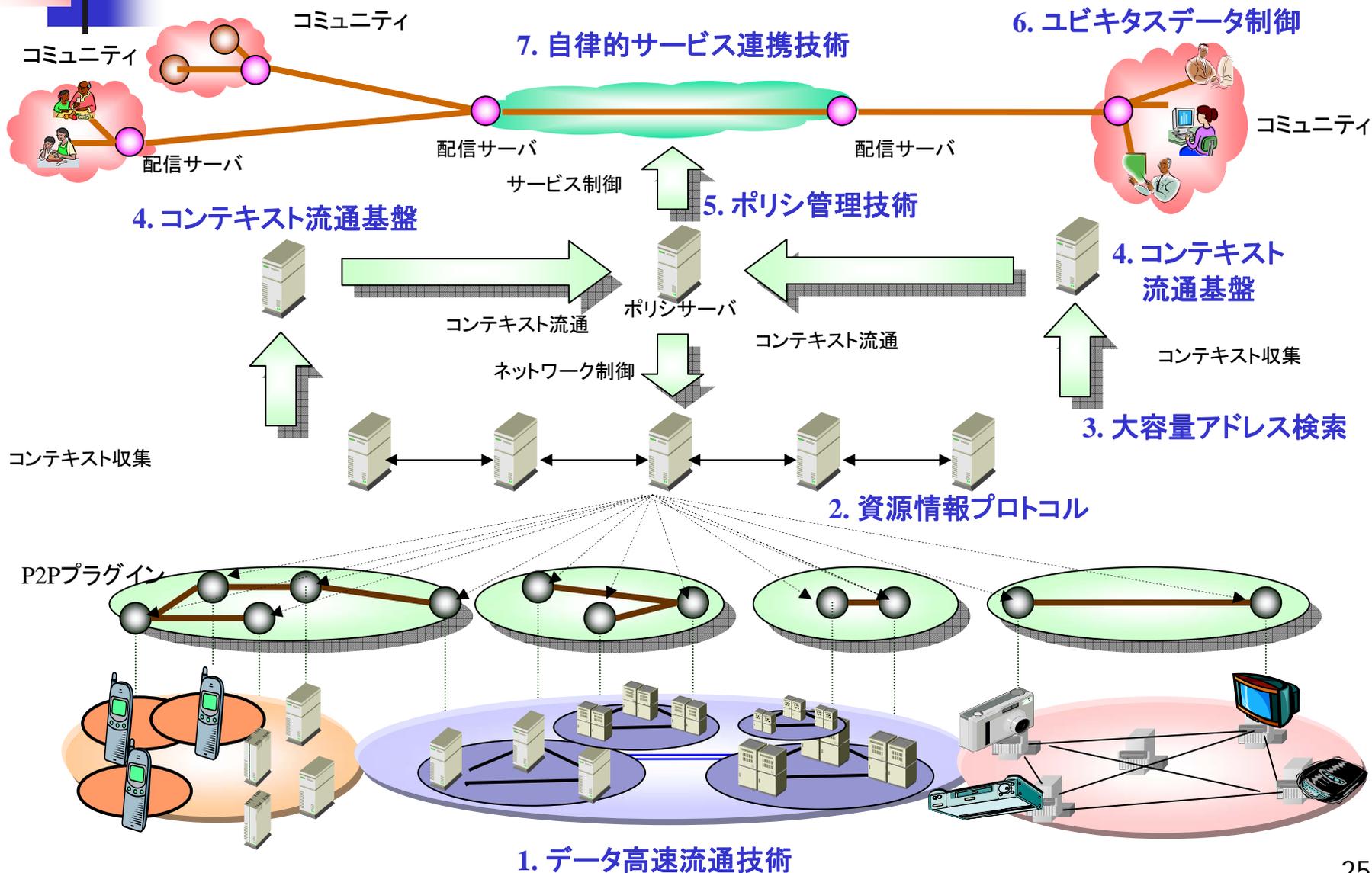
汎用連携IFを利用することにより、u-Photo ↔ 仮想ネットワークシステムとのシームレスな連携を実現
ユーザからネットワーク構築作業を隠蔽



T3: プライバシに配慮した情報利用

- プライバシを守るメカニズムの提供により、ユーザが安心してコンテキスト情報をサービスに提供できるメカニズムを提供
 - 3社が保持するプライバシー技術を補完的に組み合わせることで、プライバシー保護機能を強化
- NECのプライバシー保護技術
 - コンテキストフィルタリング技術－状況に応じて情報を変換
 - コンテキストを保護する手段として、コンテキスト変換(嘘によるコンテキスト情報の隠蔽)機能を提供
 - PUB/SUBモデルにおける、継続的なプライバシー保護に対応
 - 他社の技術
 - ID変換GW技術(富士通殿)－個人情報に対する所有者以外の関連付けを阻止
 - アクセス制御技術(KDDI殿)－設定したポリシーにより情報開示可否を判断

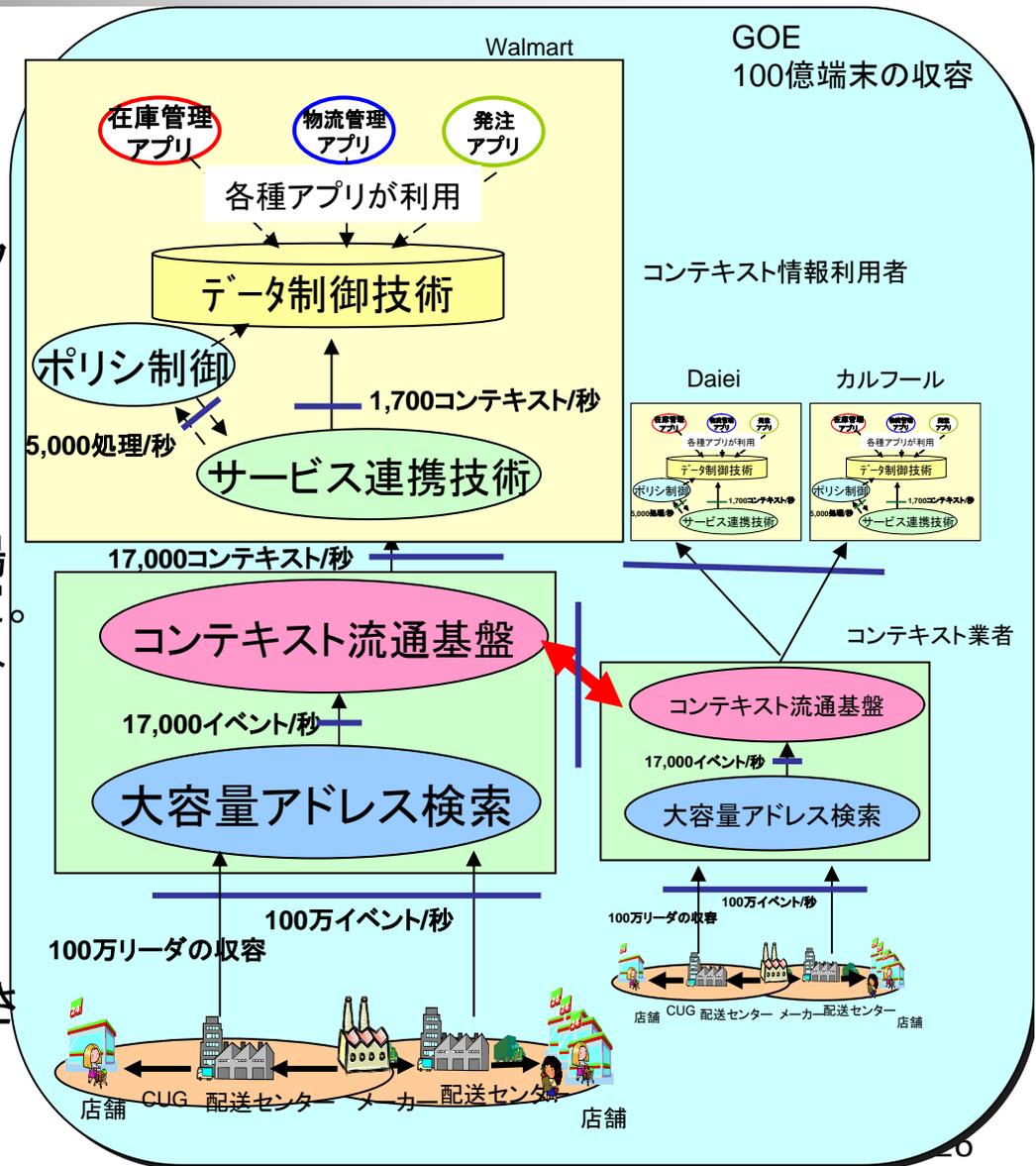
委託研究の技術詳細:目次



想定ユースケース

[RFIDトレースシステム]

- 100億個のRFIDタグを付けられた商品を管理する物流管理を想定
- ある地域内（店舗、倉庫、工場、トラックなど）にある商品群の状態をリアルタイムに管理する。
- RFIDリーダーは、店舗の棚ごと、倉庫のラックごと、トラックごとなど 100万台を想定。
- RFIDには、製品名や製造日、製造工場など100属性程度と対応していると想定。
- RFIDリーダーは、1秒ごとRFIDタグを読み取り、サーバはAPに応じて必要な統計情報を施し、データベースに記録
- データベースに記録されたデータは各種アプリから利用すると想定
- ネットワーク上には複数のコンテキスト情報利用者や複数のコンテキスト業者の端末やサーバが総計 100億台接続されていると想定



1-1. 資源情報通信プロトコル

情報家電を含む多様な機器が接続可能な仮想ネットワークを用途に合わせてオンデマンドに構築

■ 解決する課題

ネットワークの利用目的、接続端末が多様化

→ 固定的な物理ネットワークでは対応できない

■ 従来の網仮想化技術の問題点

■ 情報家電を含む多様な接続端末をサポート不可

(∵ 接続端末への専用ソフトインストールが必要)

■ 仮想網構築に手動による複雑な設定が必要

(∵ 用途に適した仮想網をユーザが設計)

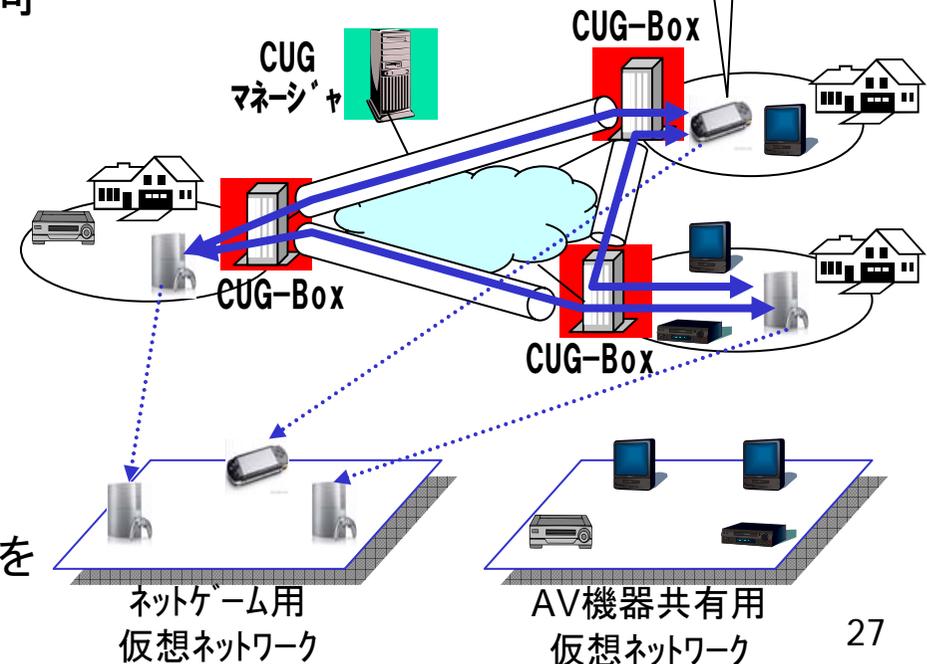
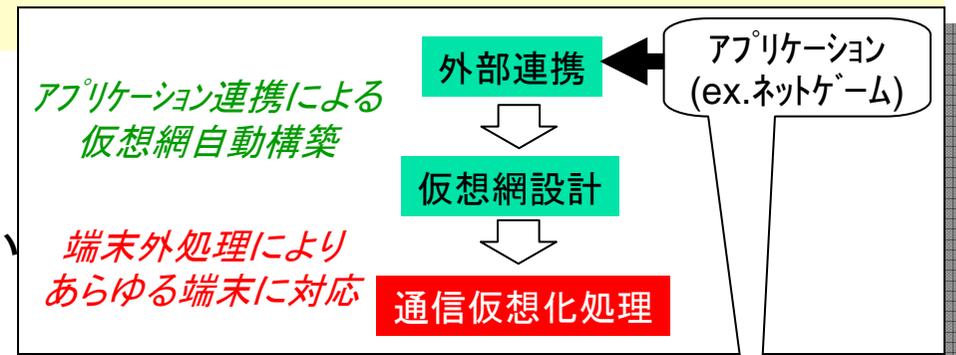
■ 新技術のアピールポイント

■ あらゆる端末が接続可能な仮想網を構築
接続端末の外部で通信仮想化処理を実行

→ 接続端末の処理能力を問わない

■ 用途毎に仮想網を自動構築

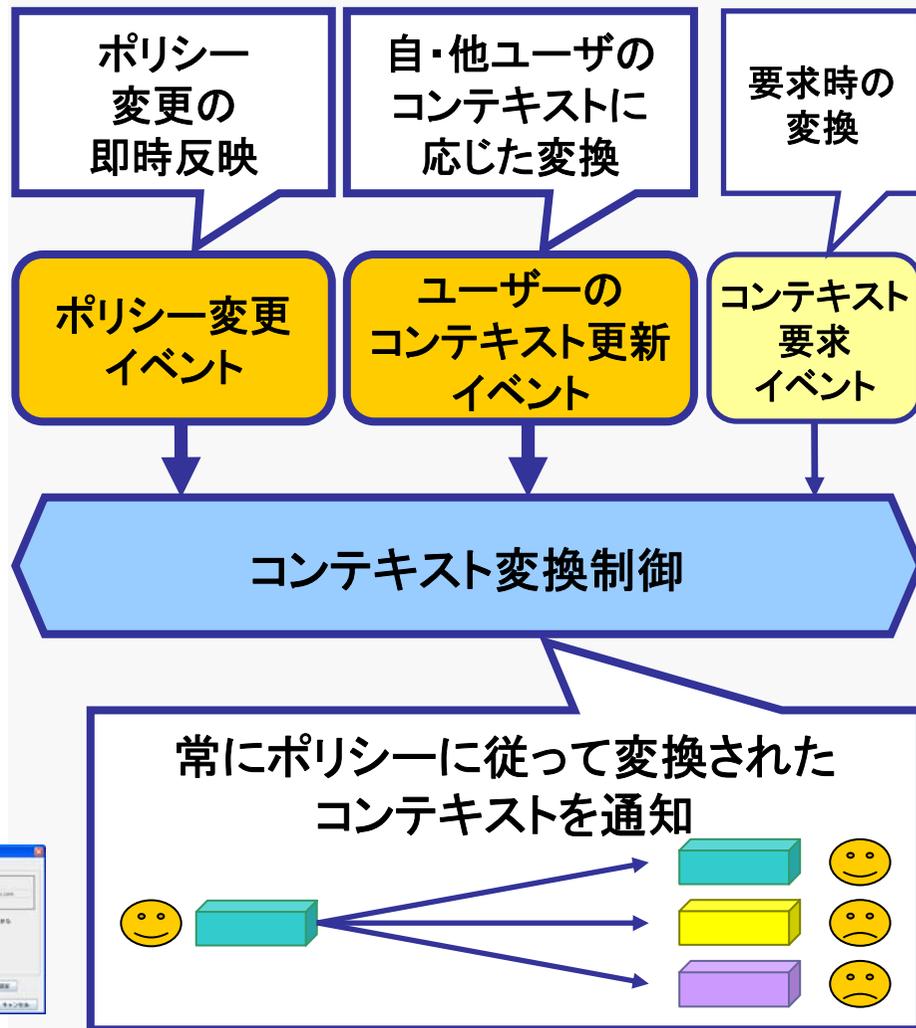
汎用連携IFによりアプリケーションと連携して仮想網を自動構築 → ネットワーク構築作業をユーザから隠蔽



1-2. コンテキスト基盤

継続的にコンテキストを交換する環境における、プライバシーの保護

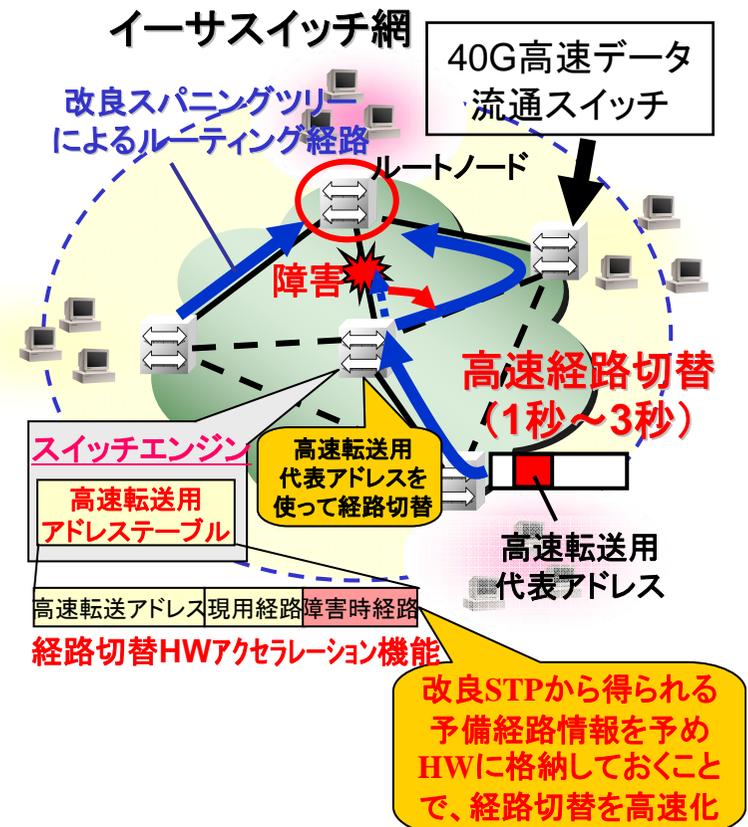
- 解決する課題
 - ポリシーに従って変換したコンテキストを通知することで、ユーザのプライバシーを保護
 - 継続的にコンテキストを交換する環境において、最新のポリシーを反映
- 従来技術の問題点
 - コンテキスト要求時にポリシーの評価をおこなうモデルでは、ポリシーの変更を即時に反映することが難しかった
- 新技術のアピールポイント
 - コンテキストを監視中の相手に対して即時にポリシーの変更を反映
 - 自・他ユーザのコンテキスト変化に応じて自動的にコンテキスト変換動作を変更
 - 動的な変更によるプライバシー情報の漏洩を回避
 - GUIによる簡単ポリシー設定



1-3. データ高速流通技術

高速転送用代表アドレスでの障害回復機能により、イーサネット上で、高速高信頼ネットワークを実現

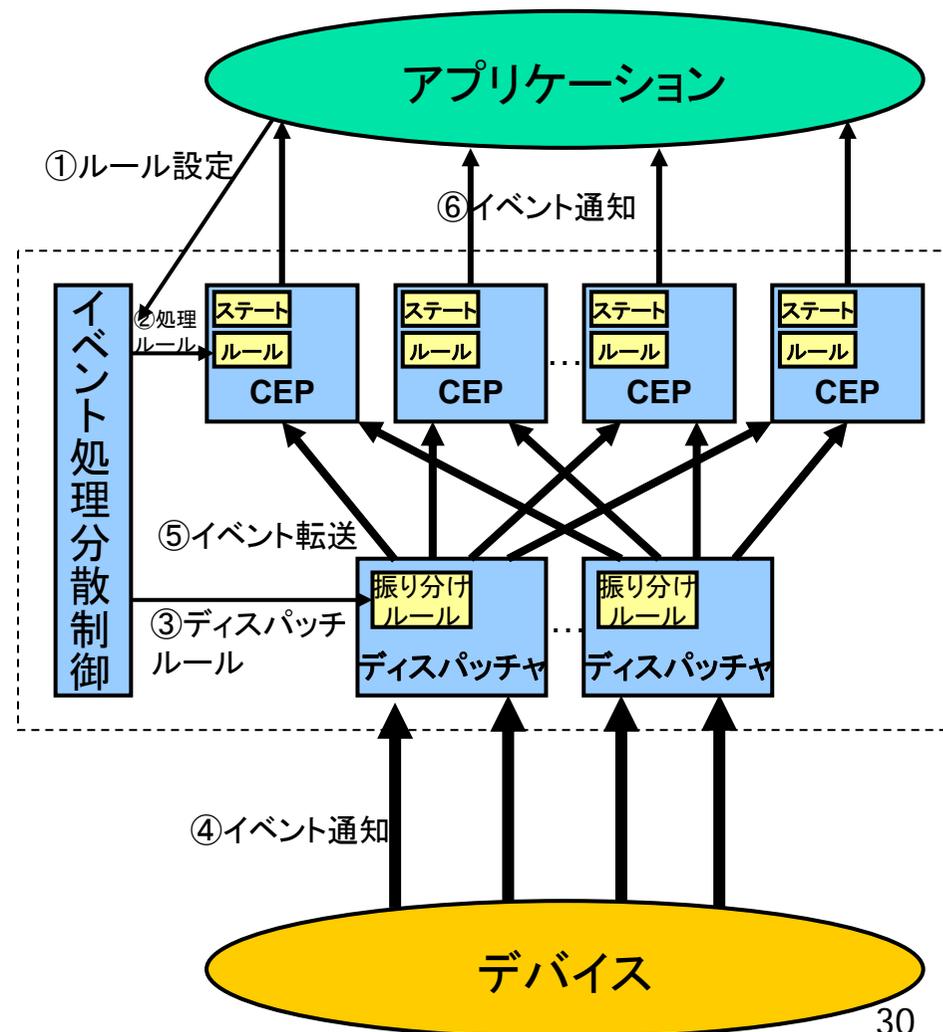
- 解決する課題
 - イーサネットにおいて、高速障害回復機能の実現
- 従来技術の問題点
 - 物理端末アドレスを用いた転送のため、障害発生時、経路情報のフラッシュと経路再学習が必要となり、大規模NWでは障害回復時間が長い(~数十秒)
 - 再学習中時ブロードキャストによるNW不安定化
- 新技術のアピールポイント
 - 高速転送用代表アドレスによる障害回復動作、経路切替HWアクセラレーション機能により、経路フラッシュと再学習を排除し、大規模NWで高速障害回復(1~3秒)
 - 障害発生時ブロードキャストの抑制によるNW高信頼化



1-4. 大容量アドレス検索技術

RFID・センサー等のデバイスから上がる大量のイベントをスケーラブルに処理

- 解決する課題
 - ユビキタス環境における大量のデバイスからのイベントを高スループットで処理
 - スケーラブルなユビキタスサービスを実現
- 従来技術の問題点
 - 従来のコンテキストウェアサービス基盤では大量のイベントを処理するのが困難であった。
- 新技術のアピールポイント
 - 複合イベント処理(CEP: Composite Event Processing)により高効率のイベントフィルタリング&アグリゲート&通知
 - ルール分配アルゴリズムによりCEPルールを複数のCEPサーバに分配し、負荷分散
 - ディスパッチャによりデバイスからのイベントをフィルタリング&振り分け



2-1. ポリシ管理技術

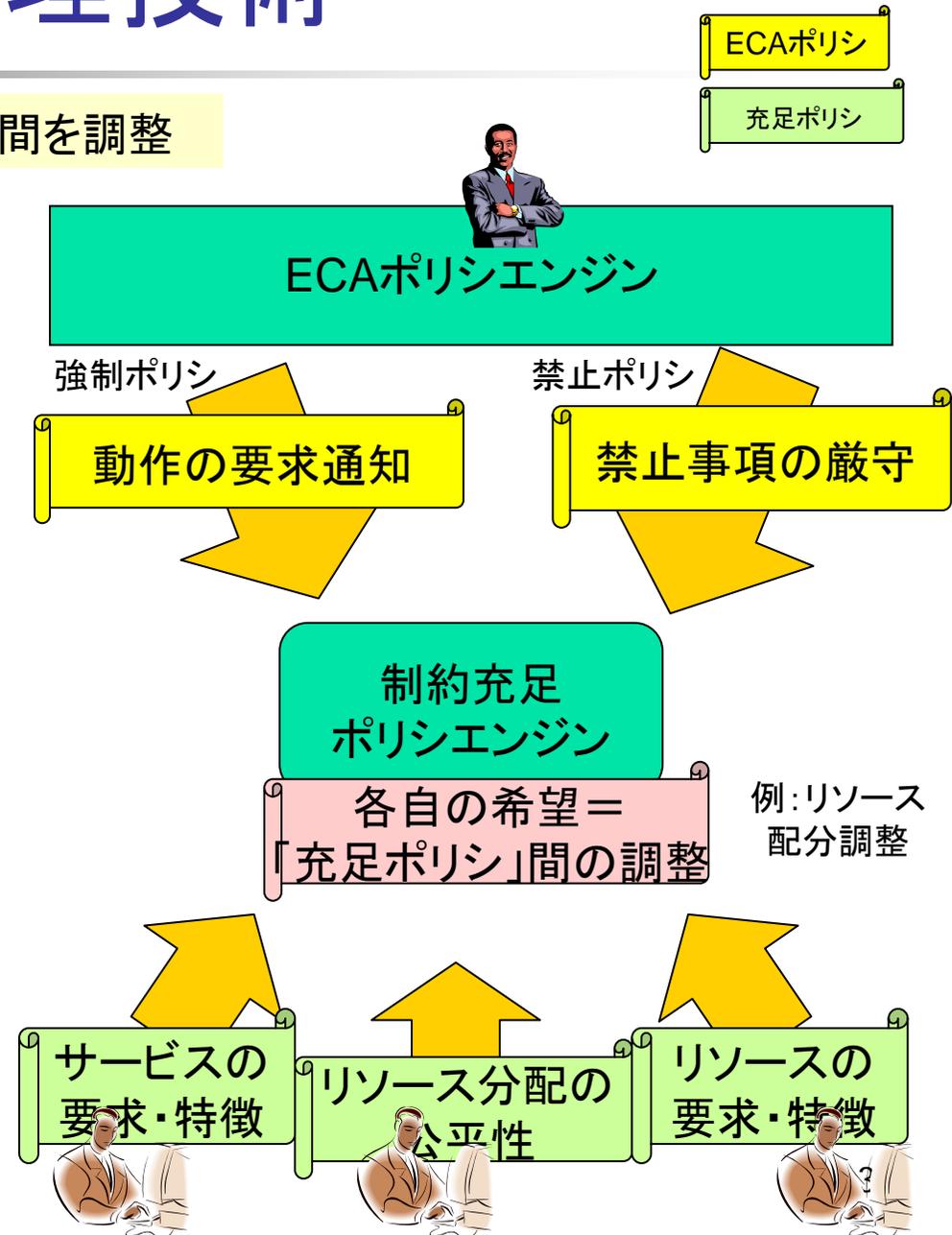
制約充足ポリシーエンジンによる管理ポリシー間を調整

【課題】

- Event-Condition-Action (ECA) ポリシによる自律管理
 - 「条件→動作」で記述するポリシ
 - システムの状態に応じて、適切な管理操作を記述可能
- 課題: 複数の管理者が ECA ポリシを設定する場合:
 - ECA ポリシ間で矛盾発生の可能性
 - 複数人が独自に入れるポリシの全てを満たす答えはない場合が多い

【アプローチ】

- 複数のポリシをなるべく満たす方法を実現する制約充足ポリシーエンジン
 - 「充足ポリシ=制約」をなるべく多く満たす組み合わせを発見する制約充足問題とモデル化し、解を発見
- ECA ポリシエンジンは、全体の動作を統括、クリティカルな状況に陥らないことを監視
 - システムが必ず守るべき基準規則を実施
 - システムを監視し、その状況に陥るオペレーションを禁止
- サービスへのリソースの配分問題を対象として検討中



ECAポリシー

充足ポリシ

2-2. 自律的サービス連携

M2M化や巨大コンテンツ化に伴う高負荷に対応するサービス負荷分散を実現

■ 課題

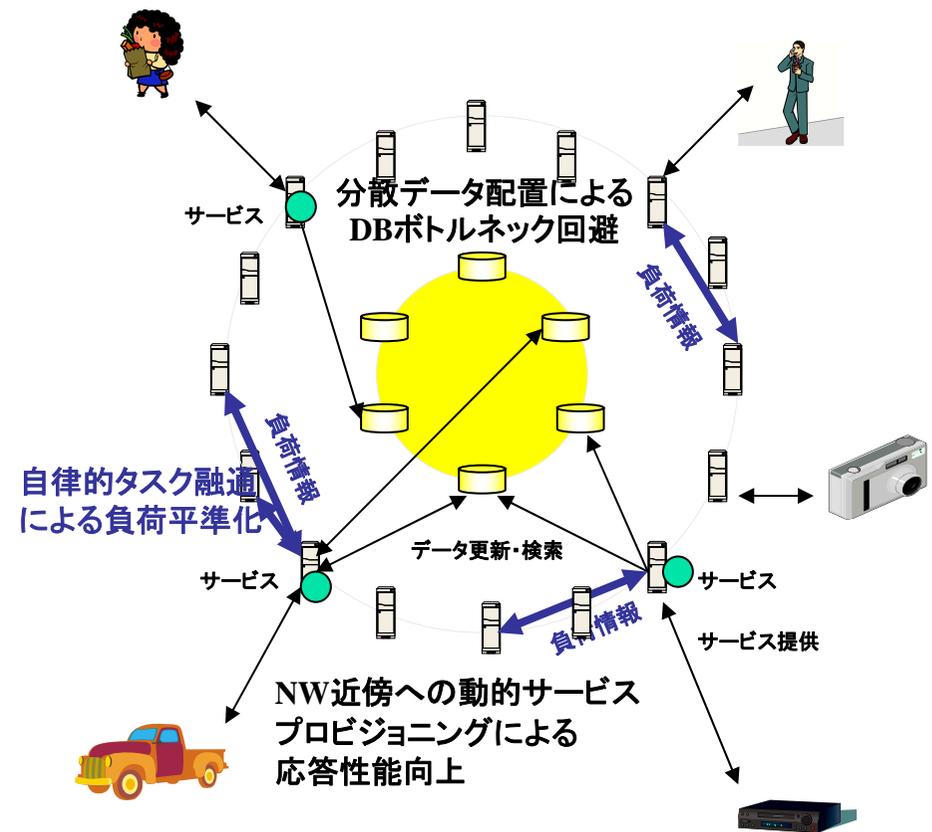
- M2M化:RFID タグやセンサーの普及
 - RFIDリーダーやセンサーから大量のリクエストが発生。
 - コンテンツの巨大化
 - 単純データ (100kB) ⇒ 音楽コンテンツ (数MB) ⇒ 映像コンテンツ (数GB)
- ⇒ 膨大な情報処理を行うための負荷分散技術が重要

■ 従来技術の課題

- ローカルロードバランサはスケーラビリティに限界
- CDNは、リードオンリコンテンツを前提に複製配備のため、更新可能なコンテンツでのDBボトルネックに対応不可。

■ 目指す技術

- ネットワーク距離を考慮したサービスプロビジョニングを自動的に行うことで、グローバルロードバランシングの管理負荷を低減。
- コンテンツ分散配置によるデータベースボトルネックの解消
- 【進捗】: 負荷分散プロトコルを試作・評価。コンテンツを配置する負荷平準化を確認。



2-3. ユビキタスデータ制御技術

膨大な数のユビキタス端末から同時にサービスアクセス要求を受けても、制御理論に基づいて最適なサーバを選択し、サービスレベル劣化を防ぐ技術。

- 解決する課題
 - バースト負荷発生時に、目標とするサービスレベルを満たすサーバ組合せの算出。
- 従来技術の問題点
 - 従来技術は、データセンタ内の全サーバが全て同一性能であることを想定していたため、ヘテロジニアスな環境には対応していなかった。
- 新技術のアピールポイント
 - 本技術では、ヘテロジニアス環境であっても、必要なサーバの種類・台数の算出が可能。
 - サーバの追加・削除が頻発する不安定状態を回避する仕組みを備える。
 - 負荷予測に基づいて事前に制御をすることで、サービスレベルの低下を予防できる。

