

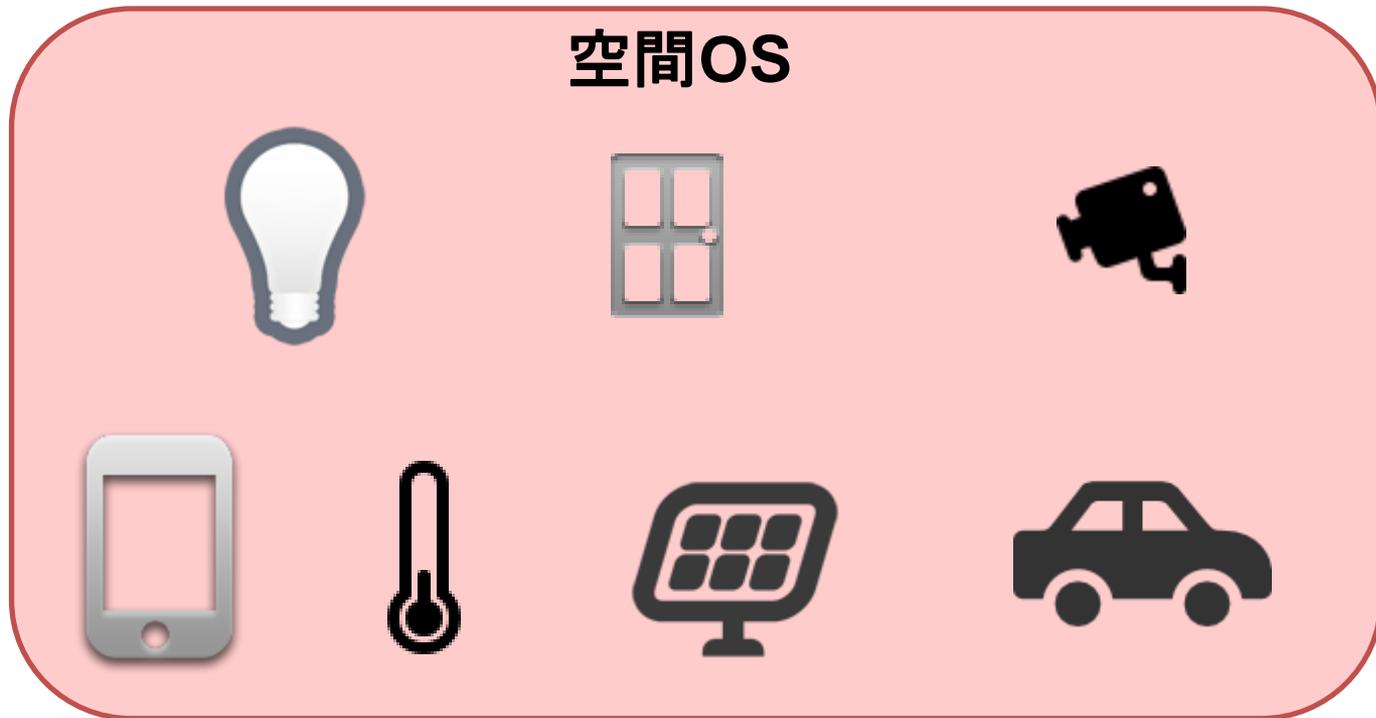
空間OSと時系列データ

2016年9月16日

先端IT活用推進コンソーシアム
ビジネスAR研究部会
日本総合システム株式会社 中川雅三

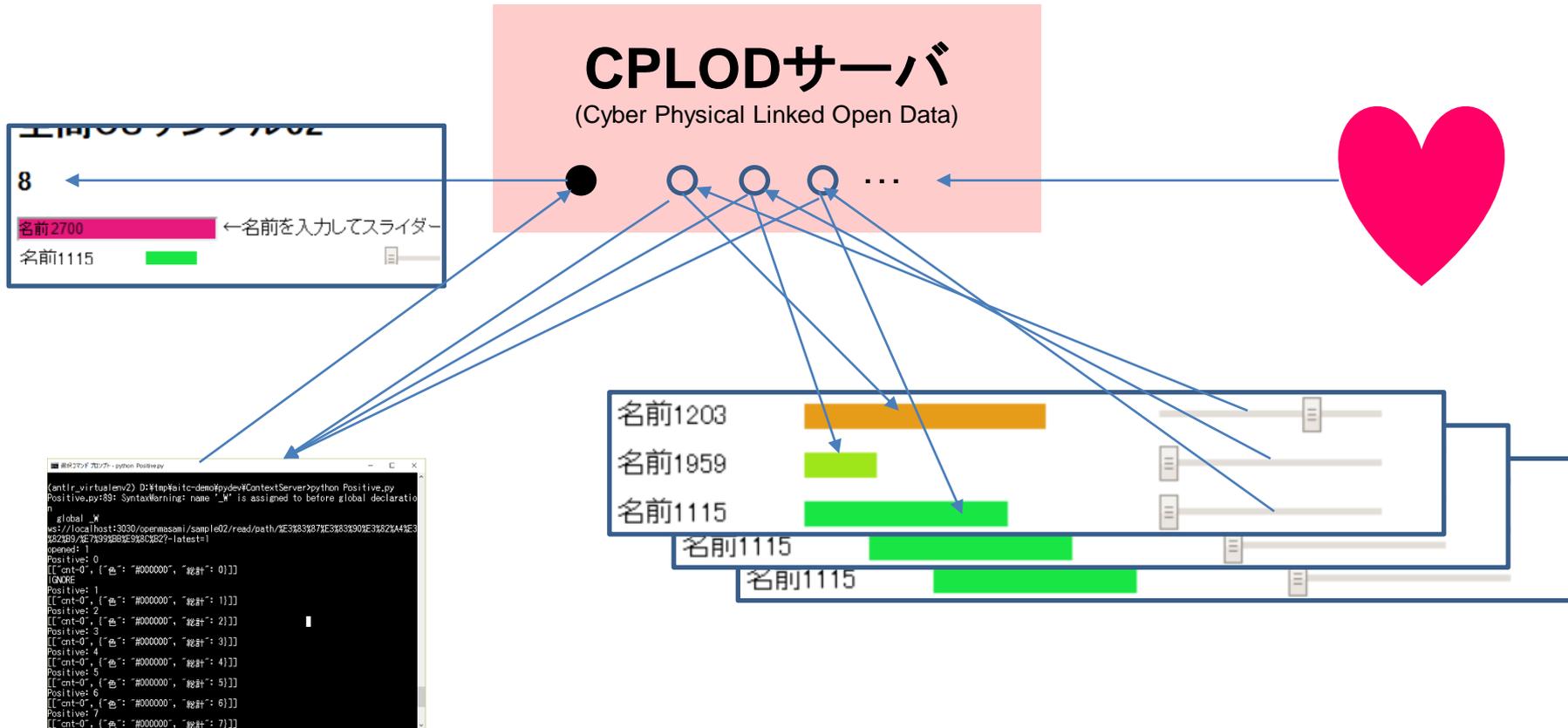
空間OSとは

- ある場所にある様々なデバイスが、データとサービスを共有しあう基盤



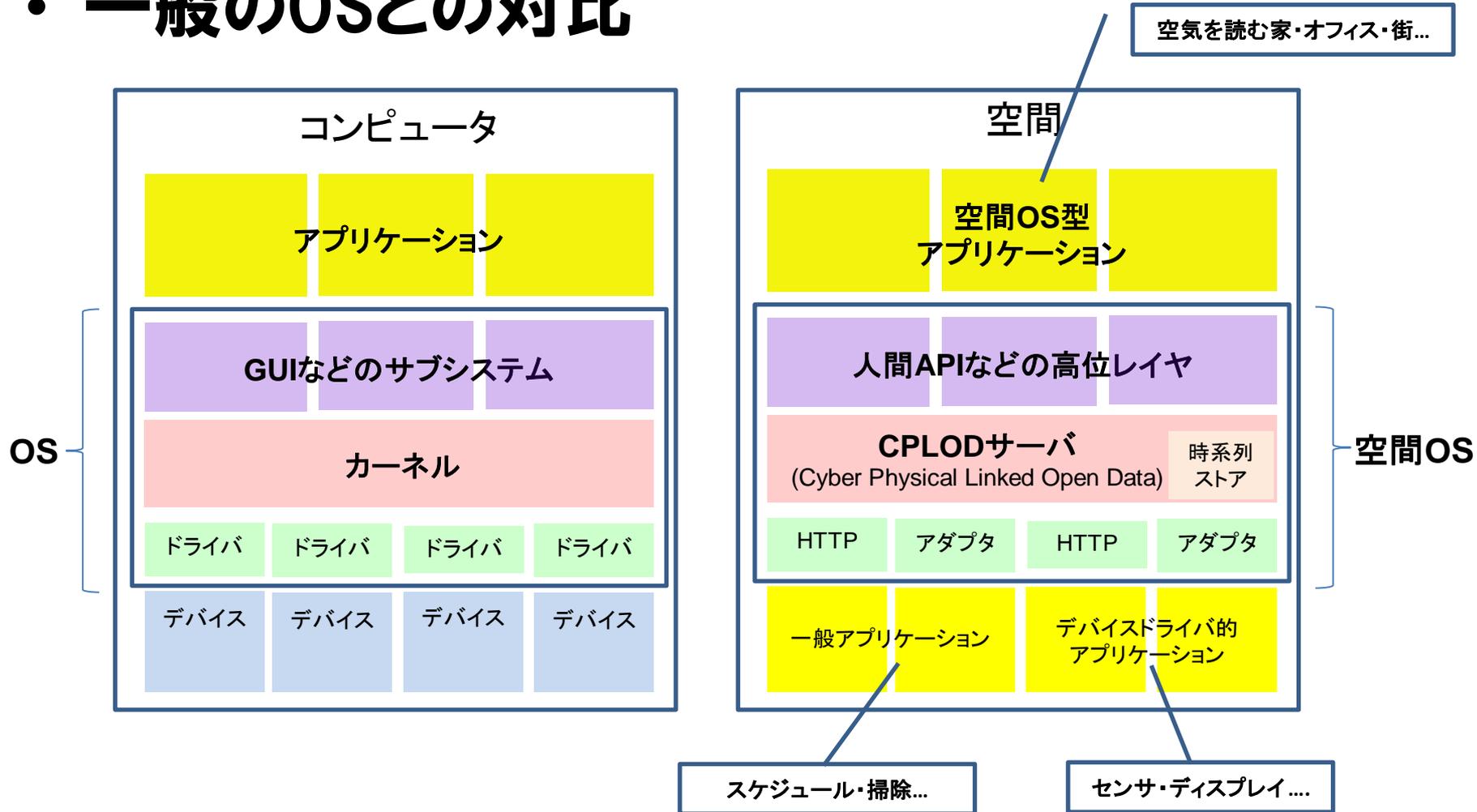
最初は「家」で実装開始

簡単なデモ



論理的な構成

• 一般のOSとの対比



- **LOD: データ共有の技術**
 - RDFでデータを表現
 - HTTPでクライアント接続
 - SPARQLで問い合わせ
- **Cyber Physical機能拡張**
 - 物理世界のデータをRDFノードへ双方向マップ
 - 物理世界の状態を即座にRDFストアへ反映
 - SPARQLで物理世界を制御
 - 変化をWebSocketで通知
 - ノードごとにアクセス権限制御

時系列データ処理

時系列データ処理の種類

- **ストリームデータ処理**
 - 受信したデータをどんどん処理
- **蓄積データ処理**
 - ストリームの再生
 - 故障・メンテナンスなどの停止から、ストリームデータ処理が復帰するとき
 - バッチ処理
 - まとめて処理することで効率化
 - ゆるいリアルタイム性の用途
 - 新しい処理方式を発明・採用して、過去に遡って分析するとき

- **ストリームデータ処理**
 - CPLODをそのまま使う
 - RASPIで総計毎秒1000～数1000イベント程度対応可能か。
 - バイナリ表現を採用すれば1ヶタ性能アップ？
 - 音声や動画は、処理して抽出した結果を記録
 - 音声や画像のストリーム自体はメタデータを仲介するだけとなるだろう。
- **蓄積データ処理**
 - CPLODへ時系列記録の再生機能を統合
 - 単純な機能： 指定したところから遡って取得 停止時のストリーム処理復帰
 - ファイルへ書き出し
 - バッチ処理
 - CPLODへ時系列DBを接続(未実装。ニーズがあれば)
 - 適当な処理系があれば、簡単に接続可能だろう

- **ストリームデータ処理**
 1. RDFを検索して処理対象を抽出
 2. WebSocketで対象データの変化を受信
- **ストリーム再生**
 1. RDFを検索して処理対象を抽出
 2. WebSocketで履歴を受信
 3. ひきつづきWebSocketで新規の変化を受信
- **記録ファイルによるバッチ処理**
 1. RDFを検索して対象ファイルのURL取得
 2. URLから過去データを取得

- **JSONで送受信**
 - **測定ポイントのID**
 - **サーバ受信時刻**
 - **証跡管理などに利用**
 - **時間精度が重要でない用途ではタイムスタンプとして使える**
 - **ユーザデータ**
 - **同時刻のデータとして記録するものを複数保持可能**
 - **データ型**
 - **スカラ 数値、文字列**
 - **配列 スカラの配列**

空気を読む家への応用

- **多様なデータに対応**

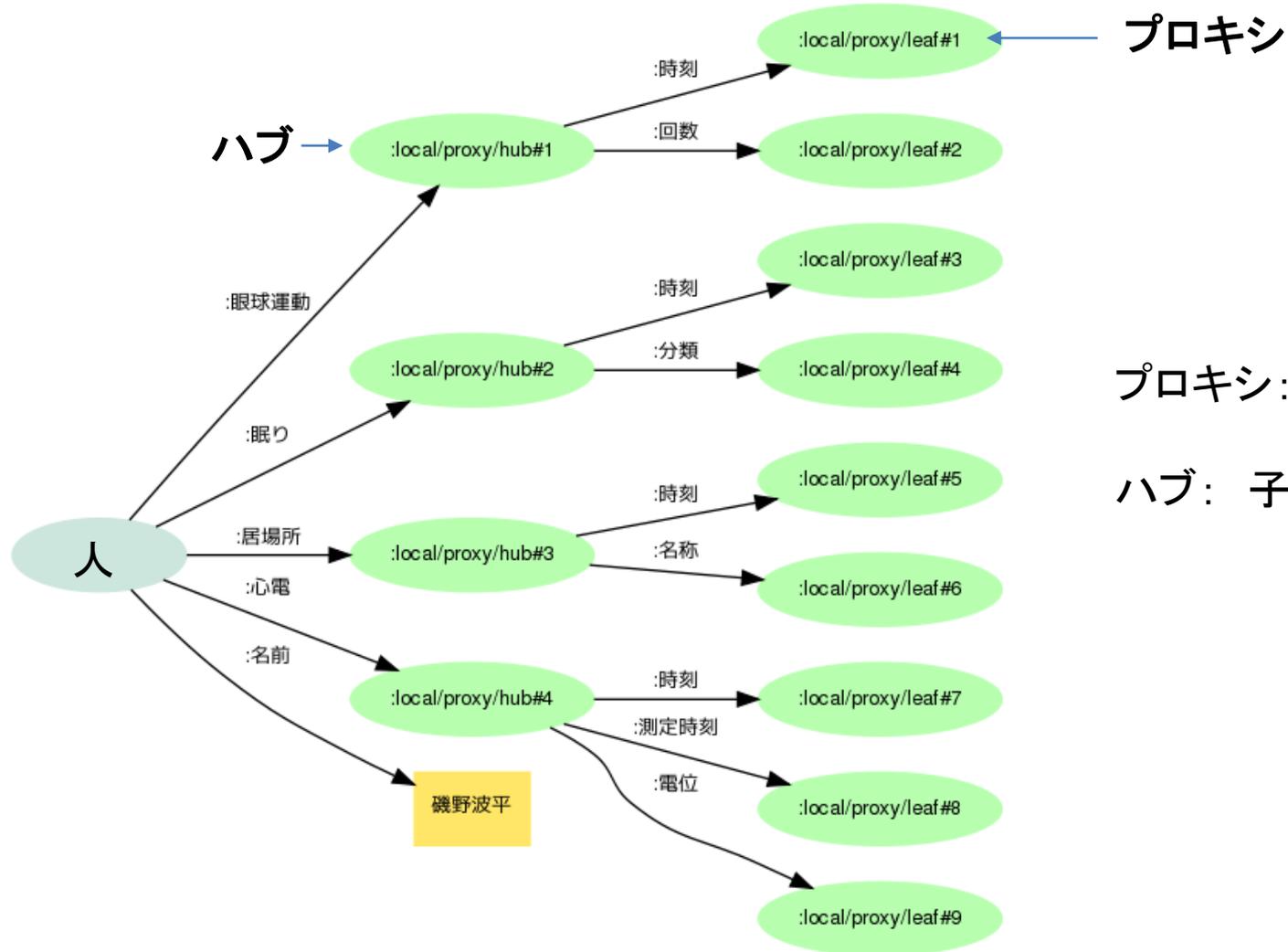
- 数値(スカラ・ベクタ): 気温、座標、眼球運動…
- 列挙値: 行動(立、歩、座、寝…)、眠り…
- コレクション: 顔検出(Aさん、Bさん…)、…

- **多様な発生パターンに対応**

- 周期的かイベントか
 - 心拍、人の移動、気分の変化
- 発生レート
 - 心電: 100Hz～目覚め: 0.001Hz～

} 区別しなくてよい

データ構造 RDFで表現



プロキシ

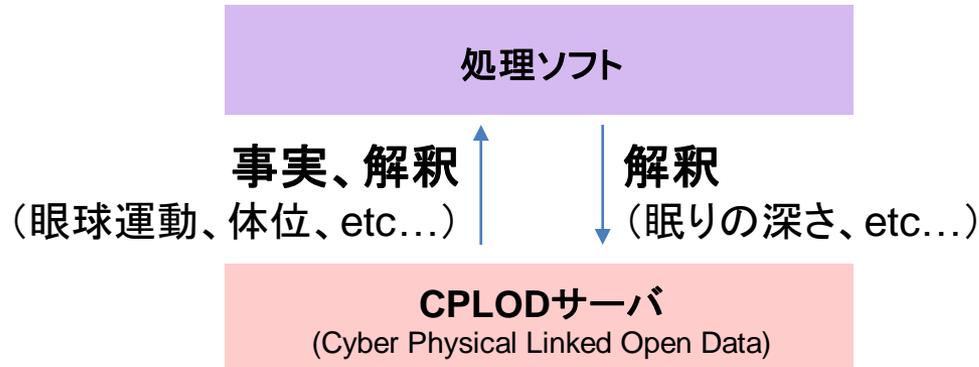
プロキシ: 測定ポイント

ハブ: 子の値を記録

- **2タイプのデータソース**
 - **事実： センサーの測定値そのもの**
 - 例： 気温、眼球運動…
 - **解釈： 事実や他の解釈から生成した値**
 - 例： 最高気温、眠りの深さ…

「解釈」の生成

- 処理ソフトが解釈して書き戻す



- 「処理ソフト」の空間OSでの位置づけ

- 空間OSアプリとして開発
- 汎用性が確認されたアプリは空間OSの構成要素へ

論理的な構成

• 一般のOSとの対比

