# Towards a Smart Data Processing and Storage Model

Work in Progresss

**Ronie Salgado**

**Inria, Univ. Lille, CNRS, Centrale Lille Lille, France**

**Marcus Denker**

**Inria, Univ. Lille, CNRS, Centrale Lille Lille, France**

**Stéphane Ducasse**

**Inria, Univ. Lille, CNRS, Centrale Lille Lille, France**

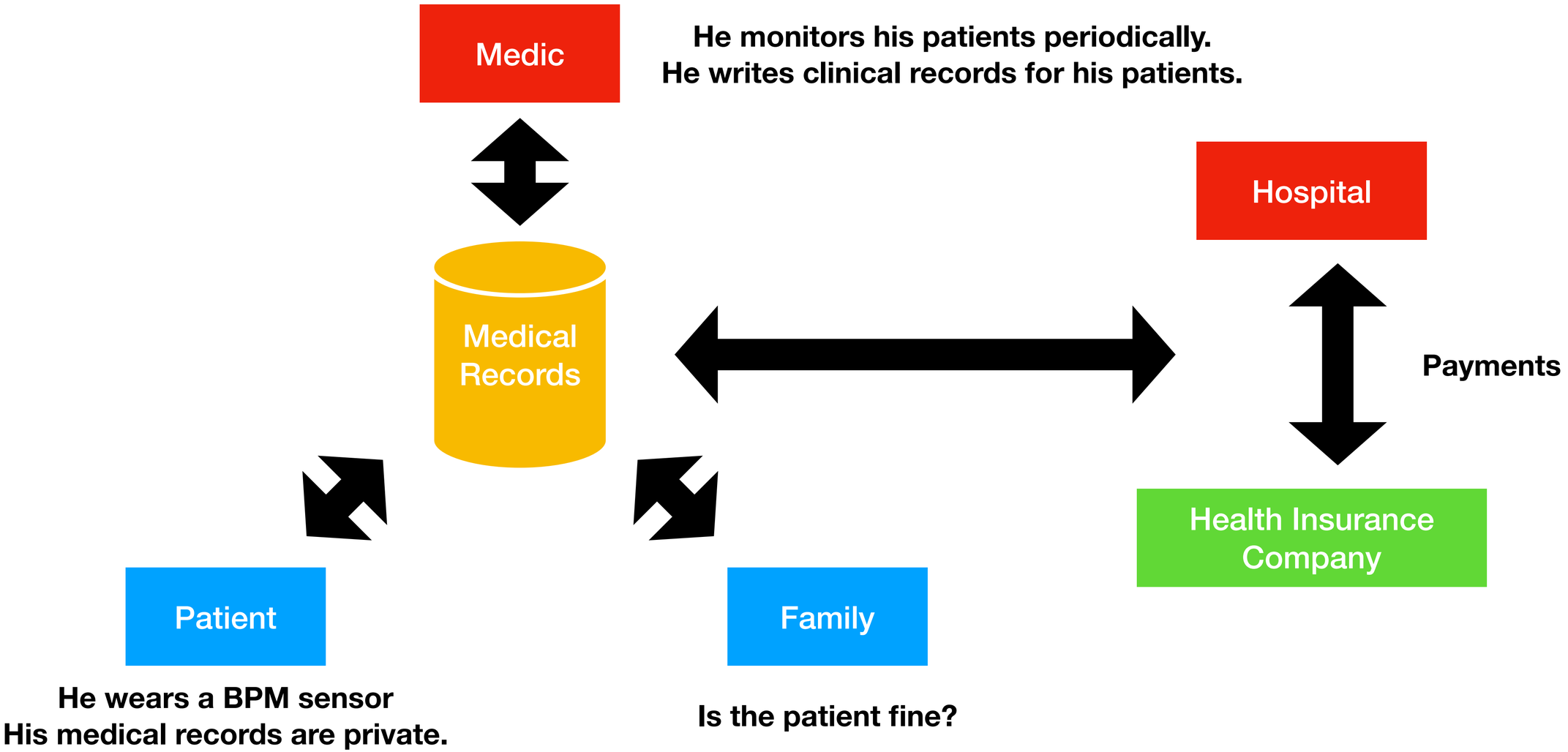
**Anne Etien**

**Université de Lille, CNRS, Inria, Centrale Lille, UMR 9189 - CRIStAL Lille, France**

**Vincent Aranega**

**Université de Lille, CNRS, Inria, Centrale Lille, UMR 9189 - CRIStAL Lille, France**

# Motivation Scenario



# Problem Statement

How do we model, design and construct a data storage and processing system that supports:

- Traceability of Origin
- Integrity Verification
- Revocation of Data

# Usage Scenarios

- Clinical Trial.
- Medical Record Keeping.
- Company Accounting.

# Traceability of Origins

We define traceability as being able to answer these questions:

- **Who** wrote this data?
- **When** this data was written?
- With **which** and **whose permissions** this data was written?

Observation: these are questions about when the data manipulation **context**. This implies:

- These questions must be **answered at the beginning** of a data manipulation context.
- We can **intercept** data **accesses** and writes **during** this context by using **Slots**.
- We call this context a **transaction**. We model them with Pharo **blocks**.

# Integrity validation

We define this process as satisfying the following properties simultaneously:

- Structural consistency: Use slots for specifying types.
- Domain defined constraints: Use assertions that must hold on the data.
- Bit-level correctness: Use a canonical data encoding to use checksums, hashes and digital signatures.

To simplify this aspect we want to use **immutable data structures** as much as possible.

# Revocation of data

Two forms of revoking data:

- **Invalidation:** the old version **can** be **accessed** (e.g. invoice correction) but is **marked as invalid**.

- **Destruction**: the old version **cannot** be **accessed** (e.g. private data).

This is a difficult problem that we have not yet solved.

# Limitations

- Revocation is not yet supported.
- Only an in-image data store is supported.
- Missing support for distributed transactions.

# Conclusions and Future Work

- We presented a mechanism for modeling and storing data with **traceability of origins**, and **integrity validation**.

- We described the **issues** of implementing **data revocability**. We want to explore the following two strategies for implementing it in an append-only store:

  - Tag invalid data by appending metadata (Invalidation).

  - Encrypt the destroyable data and forget the keys permanently when needed (Destruction).

- We want to support transactions across different nodes. We want to start supporting distribution. This is a Work in Progress.