The Role of MonetDB in the TELEIOS Project…

…and how it can be ported to your project

Stefan Manegold, **Holger Pirk**, Milena Ivanova, Ying Zhang

and Martin Kersten

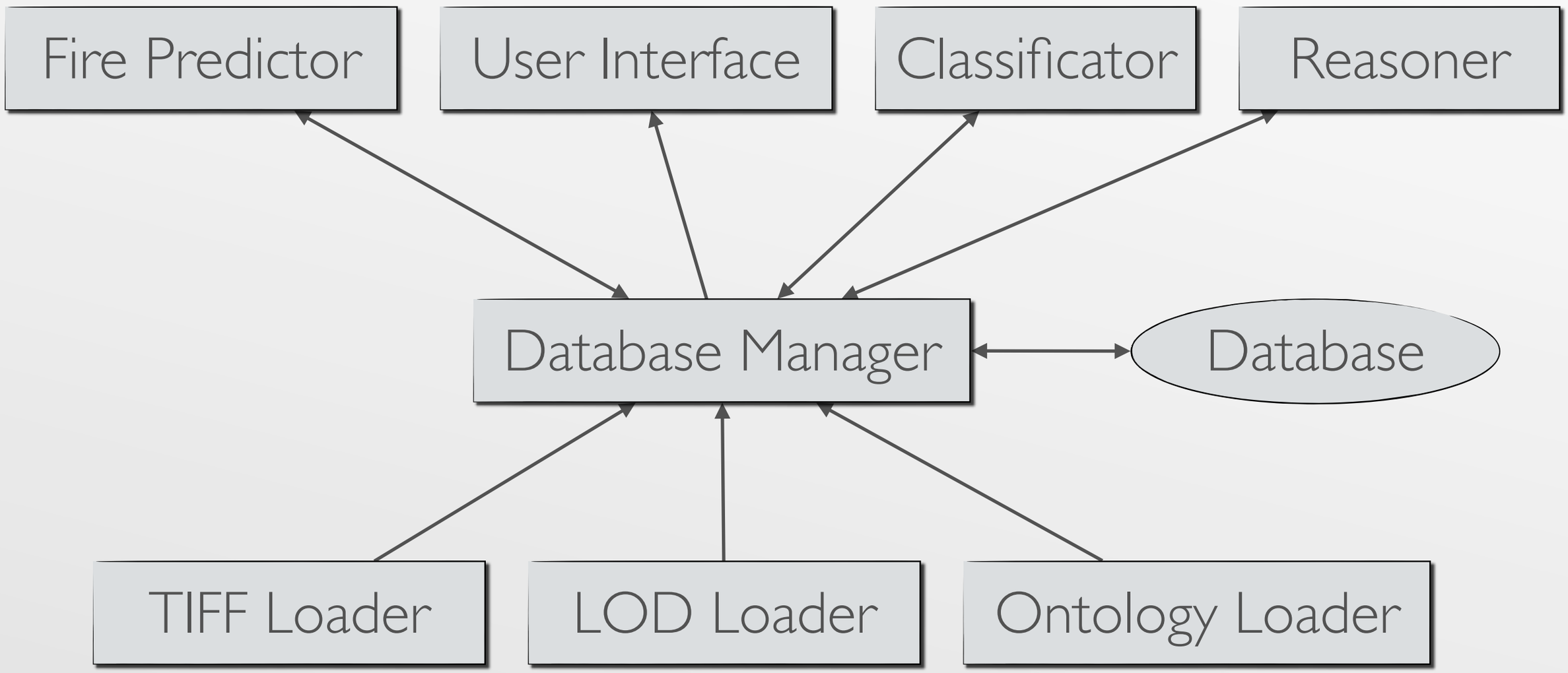
monetdb CWI TELEIOS

Requirements

- As an expert user I want to have the possibility to **split the system in different parts** and to **generate my own system** by using these parts and other modules of my own.

- As an expert user, I want the system to **maintain a set of threshold values** for fire detection that are parameterized by location, sensor and the underlying land cover so that the accuracy of fire detection can be improved.

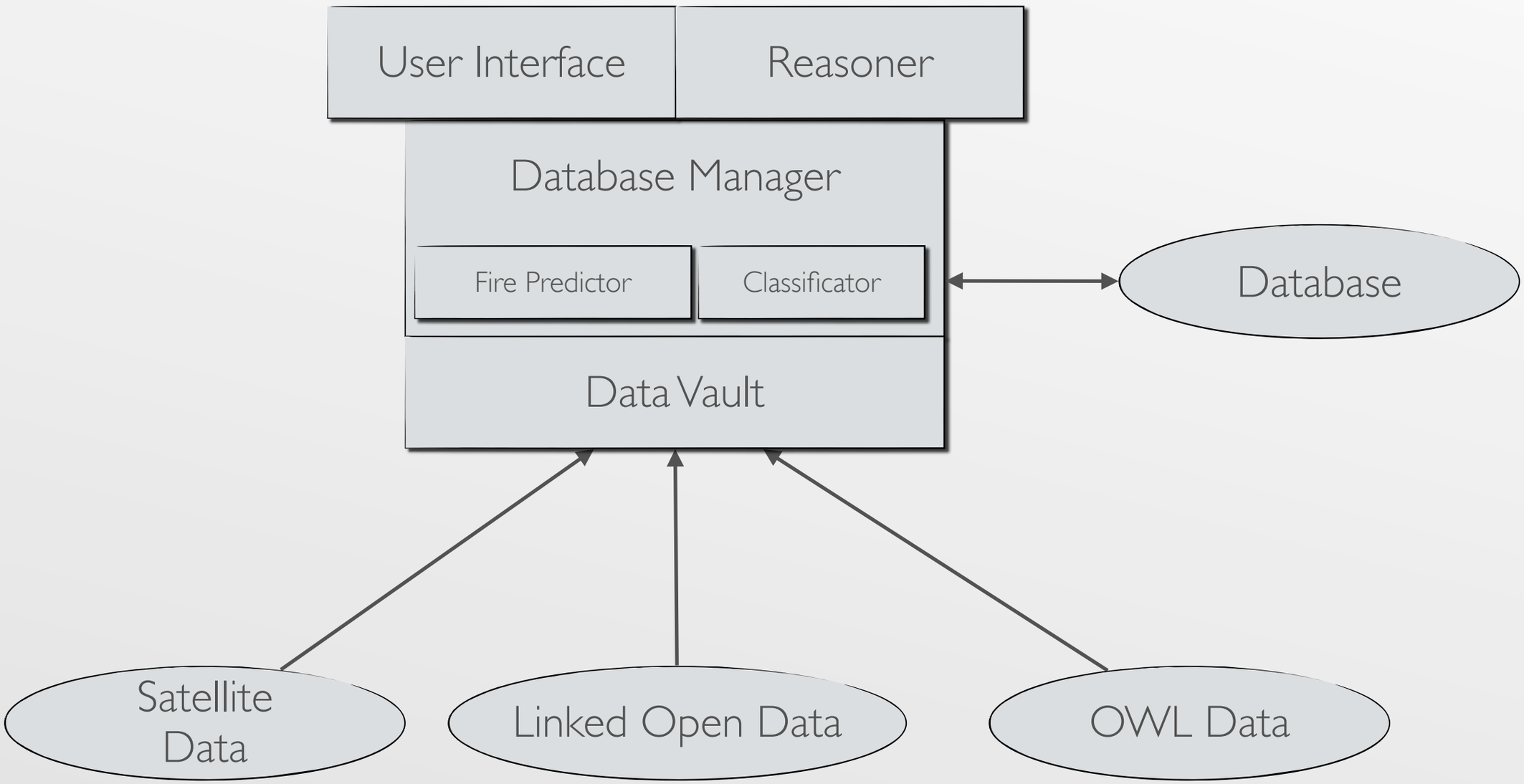
The TELEIOS Architecture (Textbook)



MonetDB Design Philosophies

- Data Management is more than ACID
- MonetDB is
  - An **easy-to-extend** data management solution
  - Aiming for the best **performance** possible
  - Targeting **Real-Life** Data Management Challenges
  - Available as **Open Source**

The TELEIOS Architecture (MonetDB Vision)



The Data Vault Framework

- Provides easy and performant integration of external data
- Just-in-Time Loading
- Relational Query Optimization
- Caching, Intermediate Recycling, View Management, ...

The Data Vault Framework

- Ongoing Research and Development
- Currently supports (Geo)Tiff, MSEED, XRIT and FITS data
- A Framework for user-supplied loaders is in development
- Loading is still user-driven (not at query time)

The Data Vault Framework

Demo

The Scientific Query Language (SciQL)

- Symbiosis of Relational- and Array-Data Processing
- The **Ease-of-use** of Python, R, ...
  - Aggregation, Slicing, Tiling, Drilling, ...
- The **Performance** of MonetDB
  - Hardware conscious, distributed, cached, ...

The Scientific Query Language (SciQL)

- Design Goals:
  - Concise expression of queries on arrays
    - Users of R, Python or C should feel comfortable
  - Maintain mindset compatibility with relational SQL
- This is a conflict that we aim to resolve (with your help)

The Scientific Query Language (SciQL)

Demo

Easy Extensibility

- Integrate existing applications MonetDB as modules
  - C, C++, Objective-C, Fortran, Java, Ada, Go, ...
- The simplicity of MonetDB provides a clean API for integration
  - Case: Porting the NOA fire detection took roughly a day

Request for Feedback

- Is this easy enough?
- Does this fit your needs?
- Do you have performance problems?
- Would you consider using a DBMS?

Request for Feedback

http://www.monetdb.org