A Decentralized Model for Information Flow Control

Andrew C. Myers and Barbara Liskov, 1997

September 23, 2015

Mikael Elkiær Christensen
michri11@student.aau.dk

Department of Computer Science
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

Introduction

Introduction 1
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

The result of this paper is a model for controlling information flow: Decentralized Label Model (DLM).

Introduction
What it is not

It is not:

Introduction
What it is not

It is not:

- Access Control

Introduction
What it is not

It is not:

- Access Control
- Authentication, Authorization, Confidentiality, Integrity.

Introduction
What it is not

It is not:

- Access Control
- Authentication, Authorization, Confidentiality, Integrity.

This means that DLM will not ensure:

Introduction
What it is not

It is not:

- Access Control
- Authentication, Authorization, Confidentiality, Integrity.

This means that DLM will not ensure:

- secure communication between applications

Introduction
What it is not

It is not:

- Access Control
- Authentication, Authorization, Confidentiality, Integrity.

This means that DLM will not ensure:

- secure communication between applications
- limited access to data once released

Introduction
What it is

It is:

Introduction
What it is

It is:

- Information Flow Control

Introduction
What it is

It is:

- Information Flow Control
- Decentralized

Introduction
What it is

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

Introduction
What it is

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

- not releasing sensitive data

Introduction
What it is

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

- not releasing sensitive data
- not implicitly releasing sensitive data

Introduction
What it is

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

- not releasing sensitive data
- not implicitly releasing sensitive data
- not giving away hints of inner workings

Introduction
How it differs

DLM differs from previous solutions as it is:

Introduction
How it differs

DLM differs from previous solutions as it is:

- decentralized

Introduction
How it differs

DLM differs from previous solutions as it is:

- decentralized
- less restrictive of allowed computations

Introduction
How it differs

DLM differs from previous solutions as it is:

- decentralized
- less restrictive of allowed computations
- not completely disallowing inter-application communication

Introduction
How it differs

DLM differs from previous solutions as it is:

- decentralized
- less restrictive of allowed computations
- not completely disallowing inter-application communication
- meant to extend current programming languages with data flow annotations

DLM Basics

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

DLM provides both static and dynamic checking of data flow.

Decentralized Label Model
Mikael Elkiær Christensen
Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

Principals represent users and other authoritative entities (e.g. groups or roles).

DLM Basics
Terminology

Principals represent users and other authoritative entities (e.g. groups or roles).

Values are entities computations can manipulate.

Department of Computer Science
Aalborg University
Denmark

DLM Basics
Terminology

Principals represent users and other authoritative entities (e.g. groups or roles).

Values are entities computations can manipulate.

Slots are value-holders (e.g. variables, objects, and other storage locations).

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

DLM Basics
Terminology

Principals represent users and other authoritative entities (e.g. groups or roles).

Values are entities computations can manipulate.

Slots are value-holders (e.g. variables, objects, and other storage locations).

Input channels are read-only sources that allow information to enter the system.

DLM Basics
Terminology

Principals represent users and other authoritative entities (e.g. groups or roles).

Values are entities computations can manipulate.

Slots are value-holders (e.g. variables, objects, and other storage locations).

Input channels are read-only sources that allow information to enter the system.

Output channels are information sinks that transmit information outside the system.

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

DLM Basics
Terminology

Principals represent users and other authoritative entities (e.g. groups or roles).

Values are entities computations can manipulate.

Slots are value-holders (e.g. variables, objects, and other storage locations).

Input channels are read-only sources that allow information to enter the system.

Output channels are information sinks that transmit information outside the system.

Labels are attached to values, slots or channels (more to follow).

Decentralized Label Model
Mikael Elkiær Christensen
Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

A label **L** is a set of owners, where each owner denotes its readers, e.g.:

$$\{o_1 : r_1, r_2; o_2 : r_2, r_3\}$$

where o_1, o_2, r_1, r_2, r_3 are principals.

Decentralized Label Model
Mikael Elkiær Christensen
Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

A label **L** is a set of owners, where each owner denotes its readers, e.g.:

$$\{o_1 : r_1, r_2; o_2 : r_2, r_3\}$$

where o_1, o_2, r_1, r_2, r_3 are principals.

The effective reader set of a label is the intersection of every reader, for **L** it is $\{r_2\}$.

Decentralized Label Model
Mikael Elkiær Christensen
Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles 8
Example
Code Example
Advanced
Conclusion
Future Works

- Labels are comparable:
  - $\mathbf{L_1} \sqsubseteq \mathbf{L_2}$ signifies that $\mathbf{L_2}$ is at least as restrictive as $\mathbf{L_1}$.

Decentralized Label Model
Mikael Elkiær Christensen
Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

- Labels are comparable:
  - $\mathbf{L_1} \sqsubseteq \mathbf{L_2}$ signifies that $\mathbf{L_2}$ is at least as restrictive as $\mathbf{L_1}$.
- Labels can be joined:
  - $\mathbf{L_1} \sqcup \mathbf{L_2}$ results in a join of owners and intersection of readers.

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

DLM Basics
Key Principles

- Labels are comparable:
  - $\mathbf{L_1} \sqsubseteq \mathbf{L_2}$ signifies that $\mathbf{L_2}$ is at least as restrictive as $\mathbf{L_1}$.
- Labels can be joined:
  - $\mathbf{L_1} \sqcup \mathbf{L_2}$ results in a join of owners and intersection of readers.
- Principals can act for other principals.

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

DLM Basics
Key Principles

- Labels are comparable:
  - $\mathbf{L_1} \sqsubseteq \mathbf{L_2}$ signifies that $\mathbf{L_2}$ is at least as restrictive as $\mathbf{L_1}$.
- Labels can be joined:
  - $\mathbf{L_1} \sqcup \mathbf{L_2}$ results in a join of owners and intersection of readers.
- Principals can act for other principals.
- Relabeling can be done, further restricting or declassifying.

Example

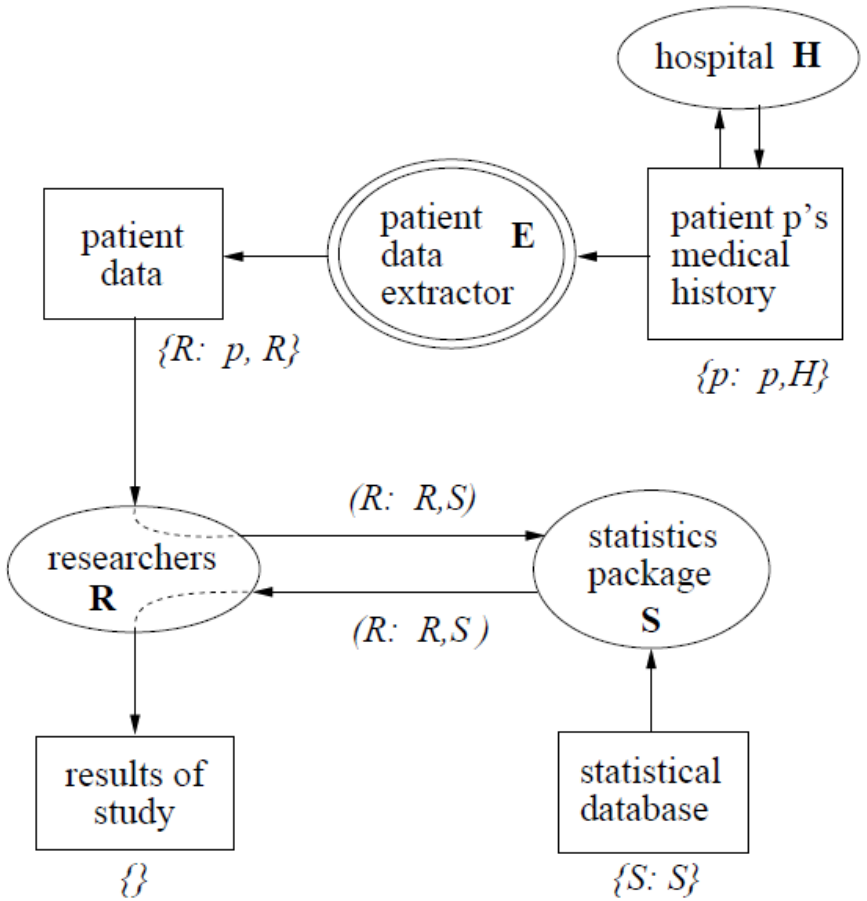

Figure 1: Medical Study Scenario

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

Example
Code Example

```
pinfo = record [ names, passwords: string{chkr: chkr} ]

check_password (db: array[pinfo{⊥}]{⊥},
                user: string {⊥},
                password: string{client: chkr})
    returns (ret: bool{client: chkr})
    % Return whether password is the password of user

    i: int {chkr: chkr} := 0                  % ⊥
    match: bool {client: chkr;                %
                 chkr: chkr} := false         % ⊥
    while i < db.length() do                  % ⊥
      if db[i].names = user &                 % ⊥
        db[i].passwords = password then       %
          match := true                       % {client: chkr;
      end                                     %   chkr: chkr}
      i := i + 1                              % ⊥
    end
    ret := false                                    % ⊥
    if_acts_for(check_password, chkr) then          % ⊥
      ret := declassify(match, {client: chkr})      % ⊥
    end
end check_password
```

Figure 6: Annotated password checker

Advanced

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

- Label polymorphism
- Run-time labels (`lb` type)
- Protected types (`protected[T]`)
- Inferred labels

Department of Computer Science
Aalborg University
Denmark

Conclusion

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

- Decentralized Label Model

Conclusion

- Decentralized Label Model
- Control of information flow

Conclusion

- Decentralized Label Model
- Control of information flow
- Static and dynamic label checking

Conclusion

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

- Decentralized Label Model
- Control of information flow
- Static and dynamic label checking
- Possible to extend existing programming languages

Conclusion
Future Works

Introduction
What it is not
What it is
How it differs
DLM Basics
Terminology
Labels
Key Principles
Example
Code Example
Advanced
Conclusion
Future Works

- Actual implementation (JIF – dead)

Department of Computer Science
Aalborg University
Denmark

Conclusion
Future Works

- Actual implementation (JIF – dead)
- Support for user-defined data abstractions

Conclusion
Future Works

- Actual implementation (JIF – dead)
- Support for user-defined data abstractions
- Formal proofs

Conclusion
Future Works

- Actual implementation (JIF – dead)
- Support for user-defined data abstractions
- Formal proofs
- Network systems

Conclusion
Future Works

- Actual implementation (JIF – dead)
- Support for user-defined data abstractions
- Formal proofs
- Network systems
- Threading

Questions?

AALBORG UNIVERSITY
DENMARK