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

Decentralized Label Model

Mikael Elkiær Christensen

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

It is not:

Department of Computer Science
Aalborg University
Denmark

Department of Computer Science
Aalborg University
Denmark

It is not:

- Access Control

Decentralized Label Model
Mikael Elkiær Christensen

It is not:

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model
Mikael Elkiær Christensen

2

It is not:

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

This means that DLM will not ensure:

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

It is not:

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

This means that DLM will not ensure:

- secure communication between applications

Department of Computer Science
Aalborg University
Denmark

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

It is:

Department of Computer Science
Aalborg University
Denmark

It is:

- Information Flow Control

Department of Computer Science
Aalborg University
Denmark

It is:

- Information Flow Control
- Decentralized

Department of Computer Science
Aalborg University
Denmark

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

3

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

- not releasing sensitive data

Decentralized Label Model

Mikael Elkiær Christensen

It is:

- Information Flow Control
- Decentralized

This means that DLM will help ensuring:

- not releasing sensitive data
- not implicitly releasing sensitive data

Department of Computer Science
Aalborg University
Denmark

Introduction
What it is

Decentralized Label Model
Mikael Elkiær Christensen

3

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

Department of Computer Science
Aalborg University
Denmark

DLM differs from previous solutions as it is:

Decentralized Label Model

Mikael Elkiær Christensen

DLM differs from previous solutions as it is:

- decentralized

Department of Computer Science
Aalborg University
Denmark

DLM differs from previous solutions as it is:

- decentralized
- less restrictive of allowed computations

Decentralized Label Model
Mikael Elkiær Christensen

Department of Computer Science
Aalborg University
Denmark

4

DLM differs from previous solutions as it is:

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

Decentralized Label Model

Mikael Elkiær Christensen

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

DLM provides both static and dynamic checking of data flow.

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

Department of Computer Science
Aalborg University
Denmark

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

Decentralized Label Model
Mikael Elkiær Christensen

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

Decentralized Label Model

Mikael Elkiær Christensen

6

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).

Department of Computer Science
Aalborg University
Denmark

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).

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

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.

Department of Computer Science
Aalborg University
Denmark

Department of Computer Science
Aalborg University
Denmark

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

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.

Department of Computer Science
Aalborg University
Denmark

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 $\{r_2\}$.

DLM Basics
Key Principles

Decentralized Label Model

Mikael Elkiær Christensen

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

Department of Computer Science
Aalborg University
Denmark

- 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.

- 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.

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

- 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.

Department of Computer Science
Aalborg University
Denmark

Example

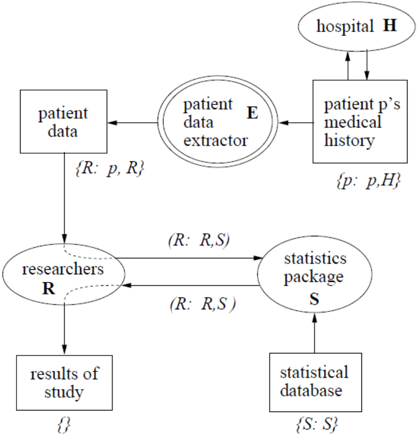

Figure 1: Medical Study Scenario

```
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

Department of Computer Science
Aalborg University
Denmark

Advanced

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

- Decentralized Label Model

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

- Decentralized Label Model
- Control of information flow

Department of Computer Science
Aalborg University
Denmark

Conclusion

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

Conclusion

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

- Actual implementation (JIF – dead)

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model
Mikael Elkiær Christensen

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

Department of Computer Science
Aalborg University
Denmark

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

Department of Computer Science
Aalborg University
Denmark

Decentralized Label Model

Mikael Elkiær Christensen

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

Department of Computer Science
Aalborg University
Denmark

Questions?

AALBORG UNIVERSITY
DENMARK