# Protecting Collaboration

**Stanford University**

**Gio Wiederhold, PI**

**Michel Bilello, Andrea Chavez, Vatsala Sarathy**

**under subcontract to**

**SRI International**

**XiaoLei Qian, PI**

# A Security Mediator for Healthcare Information

**Gio Wiederhold, PI**

**Michel Bilello, Andrea Chavez, Vatsala Sarathy**

**Stanford University**

***under subcontract to***

**XiaoLei Qian, PI**

**SRI International**

## Security: protection and assurance

**Crucial progress in protection is being made:**

**Remote Transmission**
**Authentication**
**Firewalls around domains**

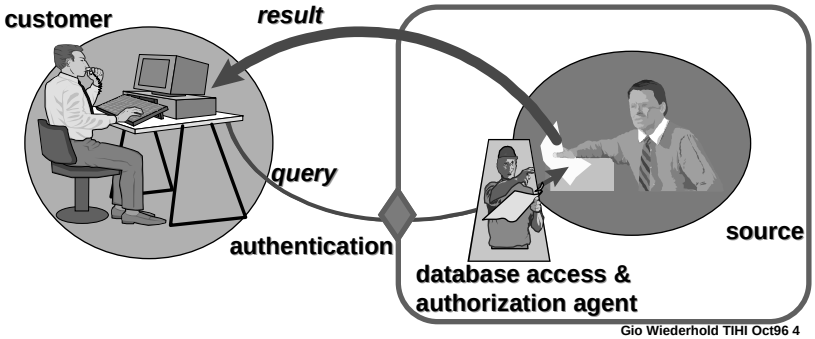
**protect against enemies.**

***Much research based on Cryptography***



## Dominant approach

- Authenticate Customer
- Validate query against database schema
- If both ok, process query and ship results



## Remaing Issue:
## Assuring Secure Collaborati

Gio Wiederhold TIHI Oct96 5

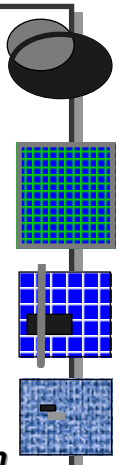
## Collaboration Needs:

- Medical Records ➔ Insurance Company
- Medical Records ➔ Medical Researchers
- Manufacturer's Specs ➔ Subcontractor
- Intelligence Data ➔ Front-line soldier

Gio Wiederhold TIHI Oct96 6

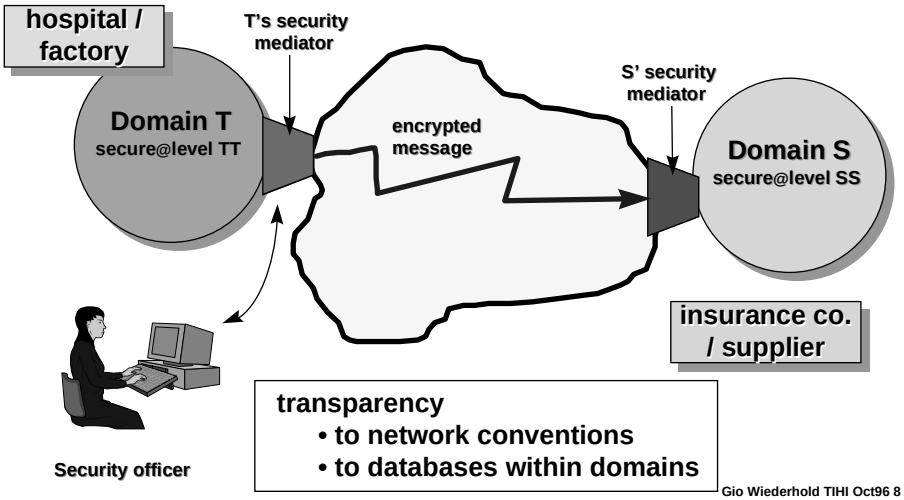
# Expected Problems

- Query can not specify object precisely
  - *Relevant history for low-weight births*
  - (helpful database gets extra stuff)
- Objects (*N*) are not organized according to all possible access classifications (*a*) = ($N^a$)
  - *Patients with heart problems, but not HIV*
- Some objects cover multiple classes
  - *Patient with stroke and HIV*
- Some objects are misfiled (can happen easily), costly/impossible to guarantee avoidance
  - *Psychiatric data in patient with alcoholism*



Gio Wiederhold TIHI Oct96 7

# Model: Interoperating Domains




transparency
- to network conventions
- to databases within domains

Gio Wiederhold TIHI Oct96 8

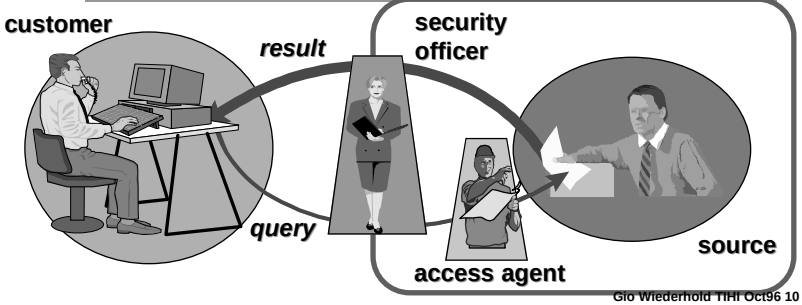
## Security Mediator

- **Software module, intermediate between "customers" and databases within firewall**
- **Resides on security's officer's machine (may have to be multi-level secure); accessed via firewall protection by customers**
- **Under control of security officer, via simple security-specific *rules***
- **Performs bidirectional screening (queries and results)**



## Agents and Privacy

**Agents help customers obtain information**
**We also need agents to protect information**
**customers must be authenticated**
**customers must be authorized to receive specific information**
**information returned must satisfy rules**



## Security Officer

- **Profile**
  - **Human responsible for database security/privacy policies**
  - **Must balance data availability vs. data security/privacy**
- **Tasks (current)**
  - **Advises staff on how to try to follow policy**
  - **Investigates violations to find, correct staff failures**
  - **Has currently no tools**
- **Tasks (with mediators)**
  - **Defines and enters policy rules in security mediator**
  - **Monitors exceptions, especially violations**
  - **Monitors operation, to obtain feedback for improvements**

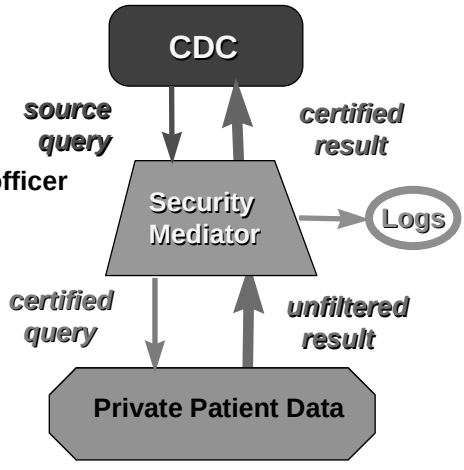


## Example: Mediation for Privacy

**Public Health Application**
- **Needs valid statistical data**
- **No access to private data**

**Security Mediator**
- **Owned by hospital security officer**
- **Screens query and result**
- **Default is Manual operation**
- **Evolves by adding rules**

**Physicians' Databases**
- **Valuable resources**
- **Need to be aggregated for significance**



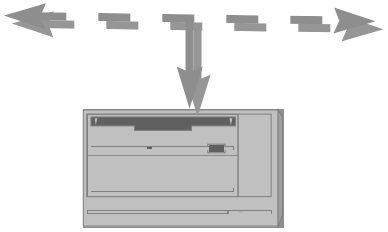
# Roles

**Security officer manages security policy,**
**not a computer specialist or database administrator.**

**Computer specialist provides tools**
**agent workstation program for security mediation**

**Healthcare institution defines policies**
**its security officer uses the program as the tool**

- **Tool provides logging for**
  - **system improvements**
  - **audit trail**
  - **accountability**
- **Formalizes *ad-hoc* practices**



Gio Wiederhold TIHI Oct96 13

# Security Officer Functions

**Security Officer**

- **defines rules for pass-through of requests**
- **any request not covered by a rule is shown**
- **shown requests can be edited and passed**

***(Passed requests are moved to the base for execution)***

- **defines rules for pass-through of results**
- **any result not covered by a rule is shown**
- **shown results can be edited and passed**

**(Passed results are moved to the top and made available)**

***All actions are logged for periodic review***

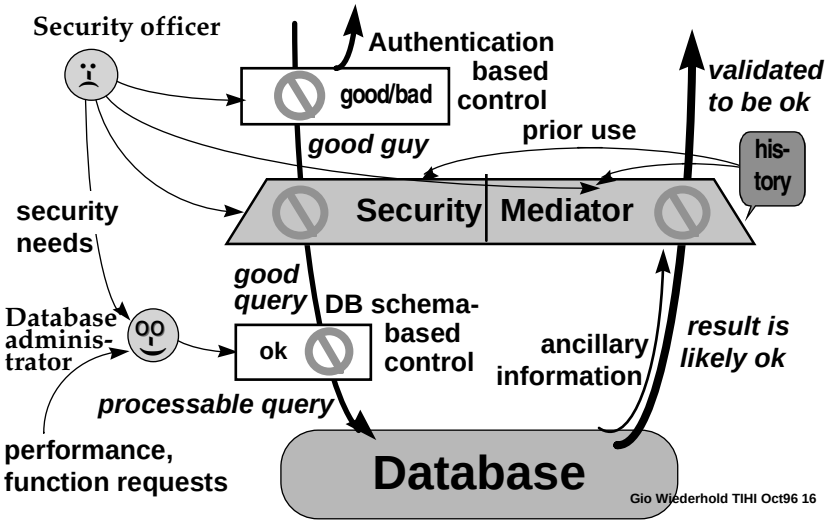
Gio Wiederhold TIHI Oct96 14

# Rules implement policy

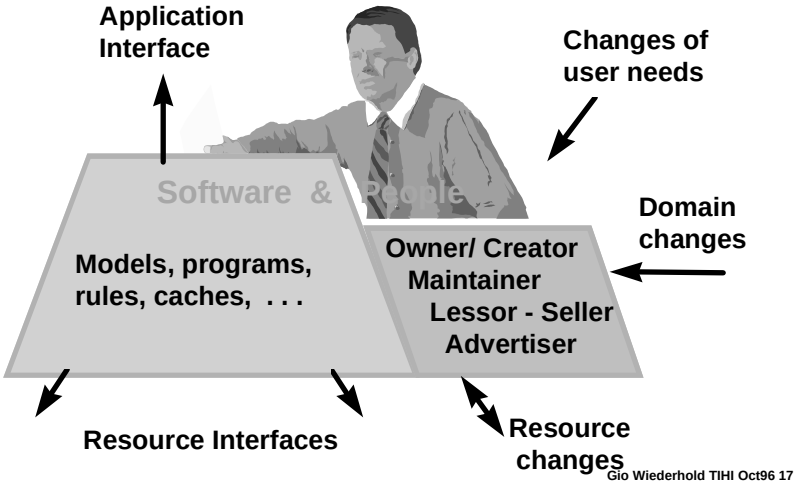
- **Tight security policy:**
  - **simple rules**
  - **many requests/responses referred to security officer**
  - **much information output denied by security officer**
  - **low risk**
  - **poor public and community physician relations**
- **Liberal but careful security policy**
  - **complex rules**
  - **few requests/responses referred to security officer**
  - **of remainder, much information output denied by security officer**
  - **low risk**
  - **good public and community physician relations**
- **Sloppy security policy**
  - **simple rules**
  - **few requests/responses referred to security officer**
  - **little information output denied by security officer**
  - **high risk**
  - **unpredictable public and community physician relations**



# Coverage of Access Paths



## A mediator is not just static software






## Agent System Differences DBA/SO

- ✪ Be helpful to customer
- ✪ Tell cust. re problems, *query may be fixed*
- ✪ Exploit DB meta-data
- ✪ Isolate transactions
- ✪ Ship result to customer

- ◐ Be helpful to security off.
- ◐ Tell sec.off. re problems, *sec.off. may contact cust.*
- ◐ Exploit customer inform.
- ◐ Use history of usage
- ◐ Ship result to sec.off. with result description (source, cardinality)

**Finding: the differences are greater than we imagined initially**

## Security Mediator Benefits

- Dedicated to security task (may be multi-level secure)
- Uses only its rules and relevant function, all directly, avoids interaction with DB views and procedures
- Maintained by responsible authority: the security officer
- Policy setting independent of database(s) and DBA(s)
- Logs just those transactions that penetrate the firewall, records attempted violations independent of DB logs*
- Systems behind firewall need not be multi-level secure
- Databases behind firewall need not be perfect

* also used for replication, recovery, warehousing



## TIHI Summary

**Collaboration is an underemphasized issue**
beyond encrypted transmits, firewalls, passwords, authentication

**There is a need for flexible, selective access to data**
without the risk of exposing related information in an enterprise

**In TIHI service is provided by the Security Mediator:**
***a rule-based gateway processor of queries and results***
under control of a security officer who implements enterprise policies

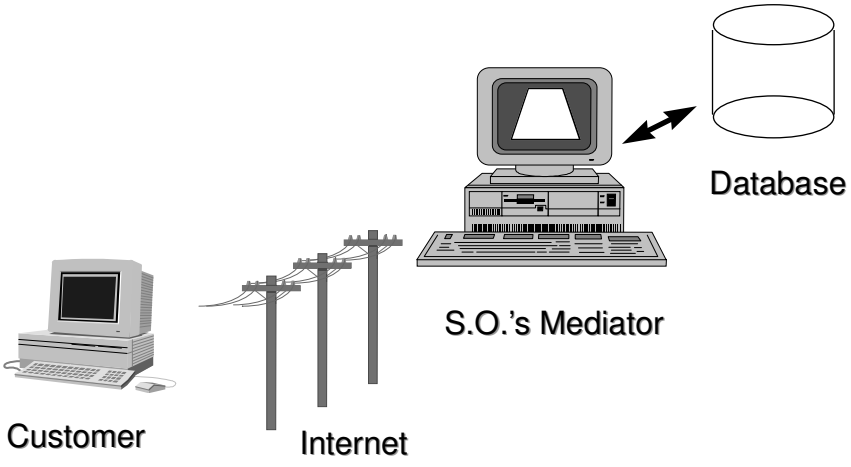
**Our solution applies not only to Healthcare**
but equally to **Collaborating (virtual) enterprises** and
in many **Military situations.**

## Backup slides

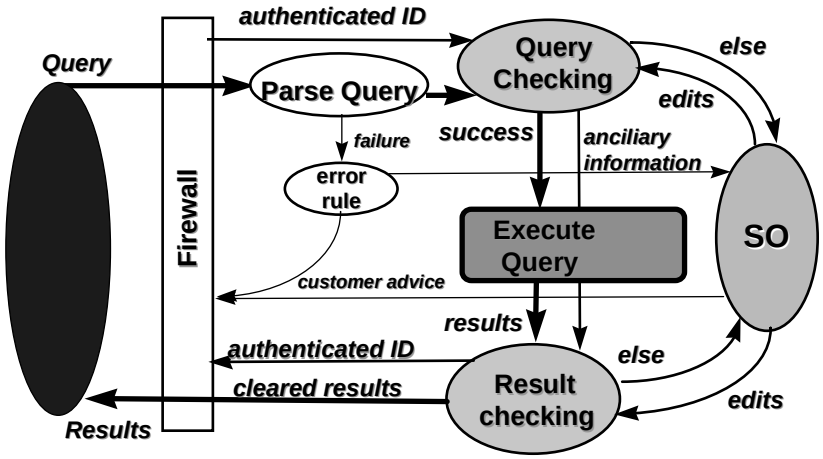
TIHI processing



## Overall Schematic

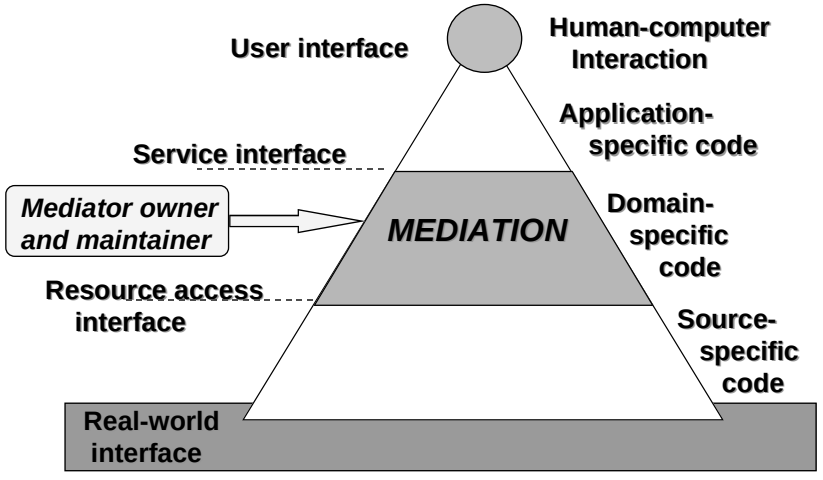


# Application of Rules




Gio Wiederhold TIHI Oct96 23

# General mediation approach: isolate value-added processing




Gio Wiederhold TIHI Oct96 24

## Rules for Security Mediators

- **Stored in a database within the mediating processor**
- **Simple but comprehensive**
- **Default is paranoid**
- **Every query is checked against every query rule (pre-processing)**
- **Every result is checked against every result rule (post-processing)**
- **Rules can be modified at all times by security officer (by experience, looking at patterns: importance of logs)**



## Rule Processing

**<u>Features:</u>**

- ***Paranoia:*** **Every applicable rule must be enforced for a query to be successful or a result to be releasable, else process by the security officer (SO)**
- ***Default:*** **If no rule applies rules then process by SO**
- **SO can pass, reject, or edit queries and results**
- **SO may inform customer, mediator will not**
- **All queries and results, successful or not, will be logged for audit**
- **Rules are stored in the mediator, with exclusive security access by the SO**

## The Rule Language

**Goals:**

- **Simple and easy to formulate by the SO**
- **Easy to enter and observe into the system**
- **Employs a collection of functions to provide comprehensive and adequate security**
- **Functions can exploit views**
- **Some functions provide text validation**
- **Some functions may need domain knowledge**



## The Rule Language - Features...

- **Rules are categorized as:**
  - **SET-UP (Maintenance)**
  - **PRE-QUERY**
  - **POST-QUERY**
- **Users are grouped into Cliques to simplify rule management**
- **Tables and their columns are grouped into segments to simplify access mgmnt**

# Rule Type Examples

- add_user user_name clique_name (Set-up)
- del_user user_name clique_name (Set-up)
- add_segment table.column segment_name (Set-up)
- del_segment table.column segment_name (Set-up)
- set_stat_only clique_name true/false (Pre)
- limit_queries_per_session x clique_name (Pre)
- limit_clique_to_segment clique_name segment_name (Pre)
- limit_min_rows_retrieved x clique_name (Post)
- limit_num_queries x segment_name (Post)
- validate_text table.column x good_words (Post)
- set_randomize_clique clique_name true/false (Post)
- set_randomize_segment segment_name true/false (Post)



# Rules... (continued)

- limit_query_intersection_clique x clique_name (Post)
- limit_query_intersection_segment x segment_name (Post)
- secure_keyword_clique keyword clique_name (Post)
- secure_keyword_segment keyword segment_name (Post)
- limit_session_time x clique_name (Pre/Post)
- limit_user_hours_end x clique_name (Post)
- limit_segment_hours_start x segment_name (Pre)
- limit_user_hours_start x clique_name (Pre)
- limit_segment_hours_end x segment_name (Post)
- limit_function function_name clique_name (Pre/Post)

# Security Table Definition

```
CREATE TABLE security_rules (
        security_function     char(32)  NOT NULL,
        object_name           char(32)   NOT NULL,
        object_value          char(32)  NOT NULL);
```

| Security Function | Object Name | Object Value |
|---|---|---|
| Limit_User | clique_name | user_name |
| Limit_Segment | segment_name | table.column |
| Stat_Only | ALL/clique | true/false |
| Queries_Per_Session | ALL/clique | integer |
| Limit_Clique_To_Segment | ALL/clique | segment_name |
| Randomize_clique | ALL/clique | true/false |
| Randomize_Segment | ALL/segment | true/false |



# Security Table Definition... (continued)

| Security Function | Object Name | Object Value |
|---|---|---|
| Validate_text | table.column | invalid_words |
| Min_Rows_Retrieved | ALL/clique | integer |
| Num_Queries_Segment | ALL/segment | integer |
| Query_Intersection_Clique | ALL/clique | integer |
| Query_Intersection_Segment | ALL/segment | integer |
| Secure_Keyword_Clique | ALL/clique | keyword |
| Secure_Keyword_Segment | ALL/segment | keyword |
| Session_Time | ALL/clique | TIME |
| User_Hours_Start | ALL/clique | start_time |
| User_Hours_End | ALL/clique | end_time |
| Segment_Hours_Start | ALL/segment | start_time |
| Segment_Hours_End | ALL/segment | end_time |
| Limit_Function_Clique | ALL/clique | function_name |

# Rule application - Overview

- **Does custumer belong to a clique?**
- **Does the customer clique satisfy all pre-query rules?**
  (e.g., Session_Start, Stat_Only, Queries_Per_session)
- **Do the columns and tables belong to a segment?**
- **Does the query satisfy all pre-query rules?**
  (e.g., valid segments)
- **Does query need re-phrasing or augmentation?**
  (e.g., Stat_Only to detailed Select)
- **Send Query to appropriate Database (or mediator)**
- **Does query result satisfy all post-query rules?**
  (e.g. Min_Rows_Retrieved, Secure_Keyword_Clique)
- **Apply any result transformation rules**
  (e.g. random falsification of data, aggregation)
- **Update log and internal statistics**



# Implementation

## Set-up

- **Security Officer enters rules into a file**
- **Rule file is parsed to generated SQL script to insert rows into the security_rules table**
- **SQL script is executed against the database**

# Implementation... (continued)

## Customer Session Loop

- **Security Mediator Workstation accepts the customer query, logs it, and passes control to the Security Mediator Software (SMS)**
- **SMS reads the security_rules table and calls many different modules (sub-routines) to validate the query (pre-query checks)**
- **If okay, SMS executes the query (Embedded SQL calls)**
- **Mediator Workstation gets results from the database and calls other SMS modules to perform the post-query checks**
- **If all checks are passed, the Mediator Workstation logs and returns results; awaits another invocation**
- **Result is accepted by customer and used or displayed**



# System Operations

- **Customer connects remotely to security officer's machine**
- **Clique membership is assessed**
- **System prompts customer for query**
- **Query is parsed and validated against rules**
- **Validated query is sent to database system**
- **Results are retrieved and validated against rules**
- **Validated results are made available to customer**