



# Database Support for Audit Trails & Intrusion Detection

---

**M.J. Atallah and S. Prabhakar**

**Students**

**Saurabh Sandhir**

**Maximillian Karpiak**

**Salvador Mandujano**

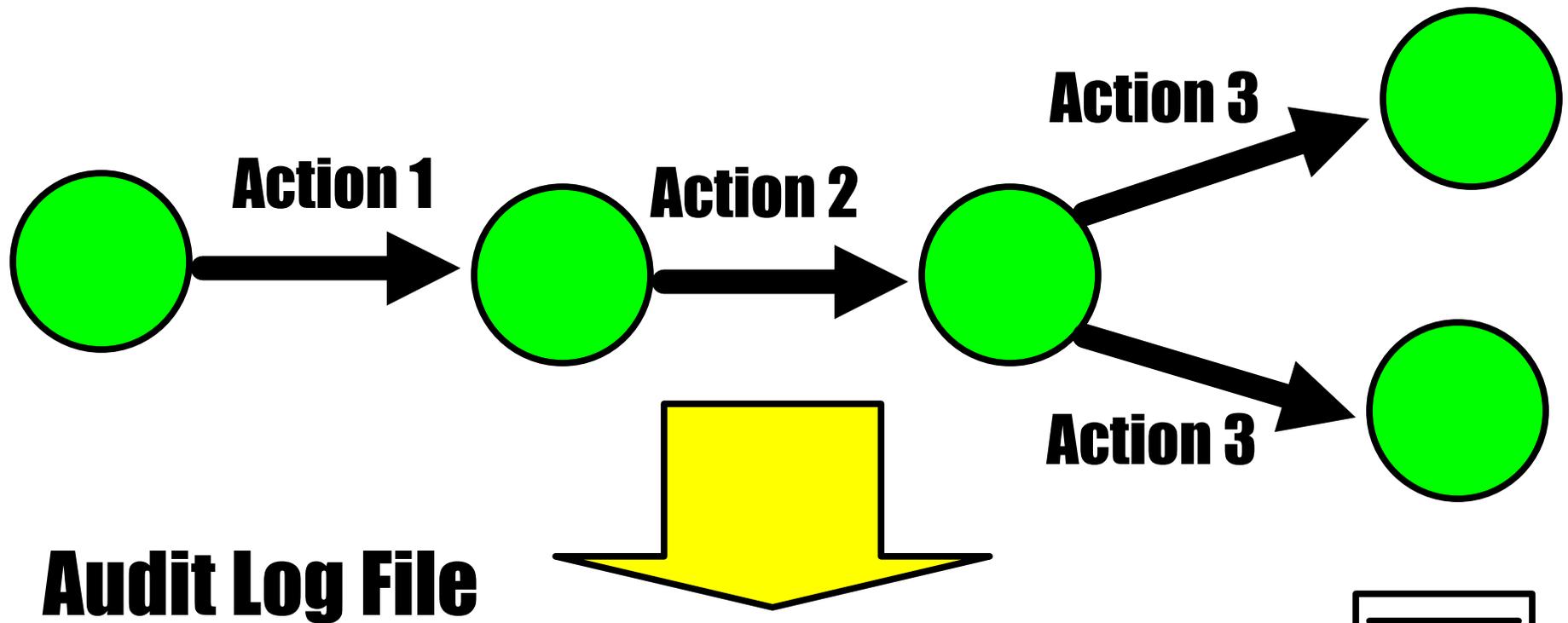


---

# Why use DB support ?

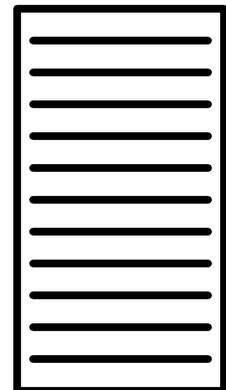
- **Flexibility and extensibility**
  - **Efficiency**
  - **Scalability**
  - **Maintainability**
  - **Ease of use and convenience**
-

# Attack and Audit Log Files



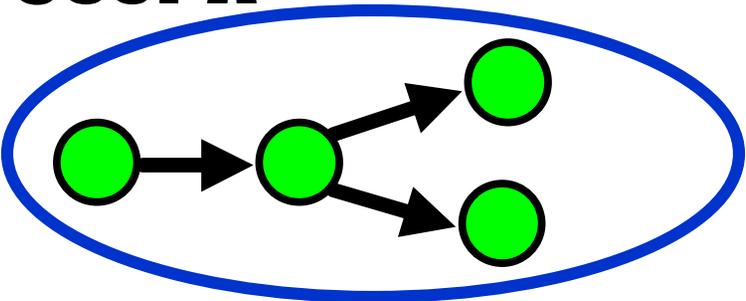
**Audit Log File**

```
Action1:user:date:time:IP address:attributes  
Action2:user:date:time:IP address:attributes  
Action3:user:date:time:IP address:attributes  
...
```

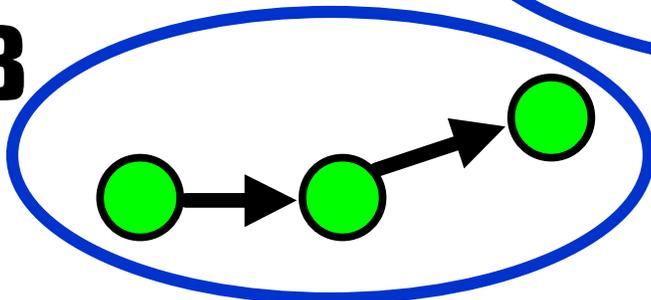


# Multiple users

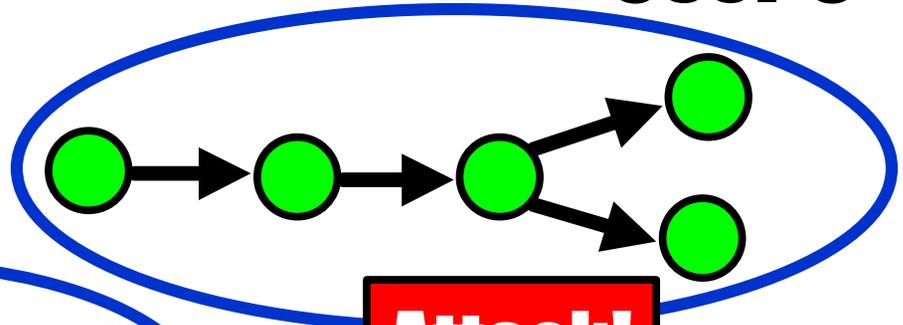
User A



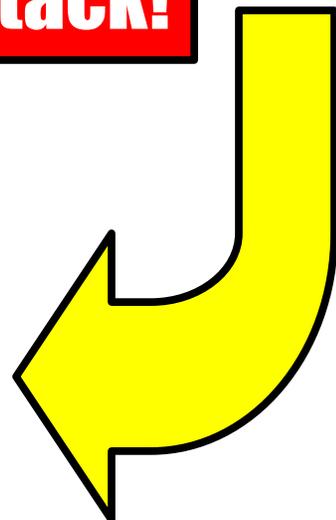
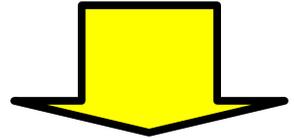
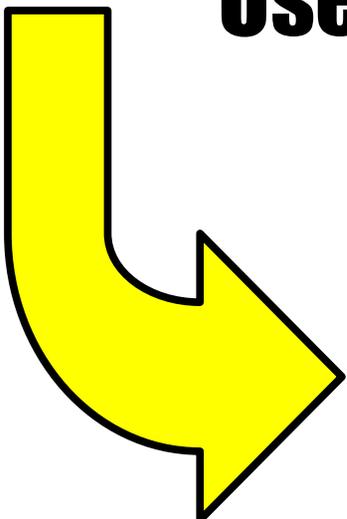
User B



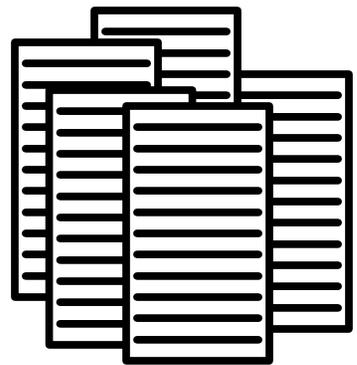
User C



Attack!

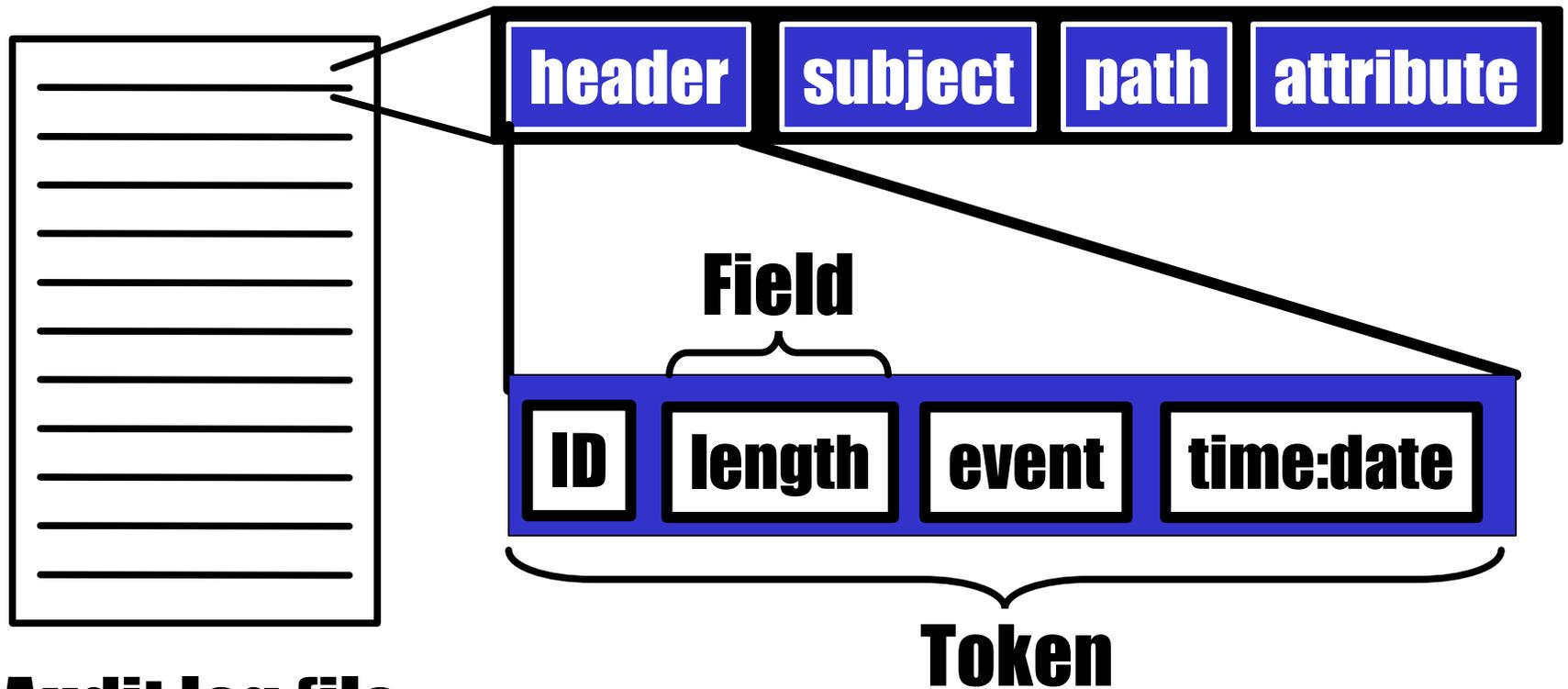


Audit Log Files



# BSM audit format

## Audit record

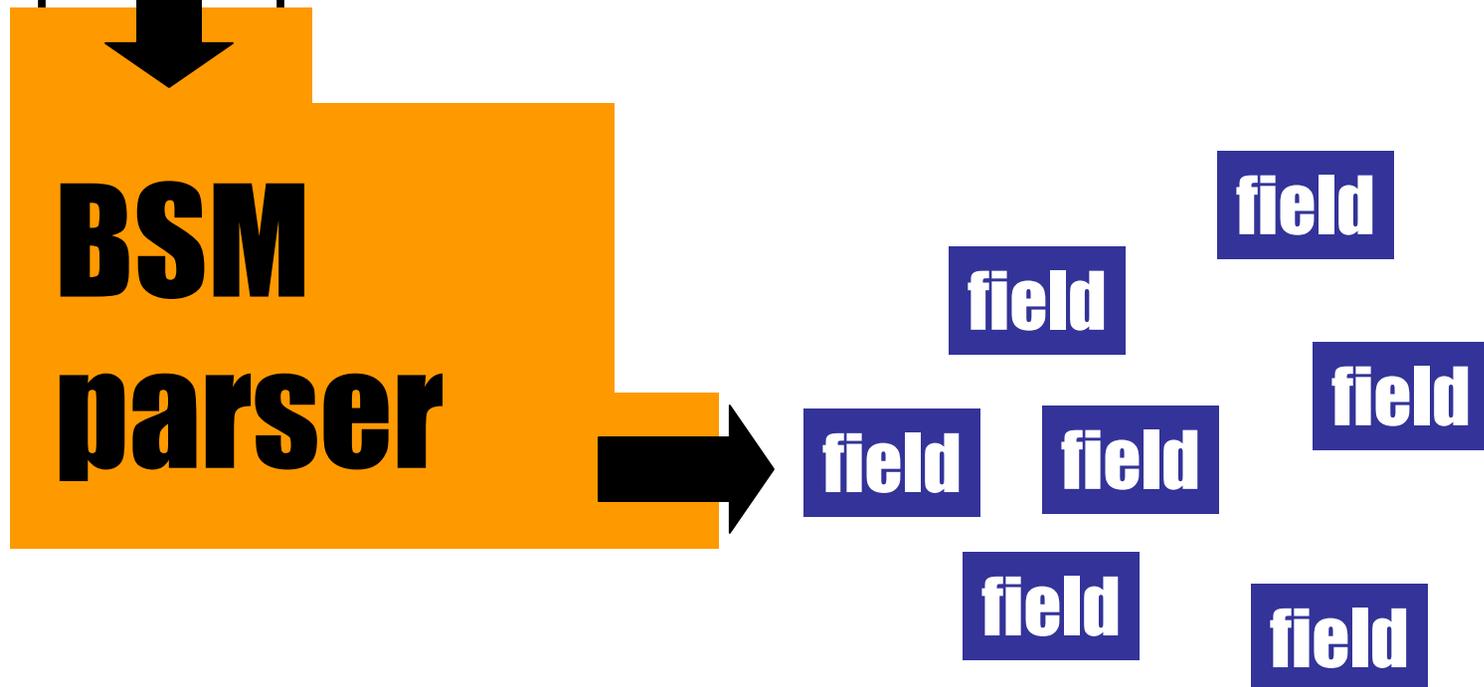


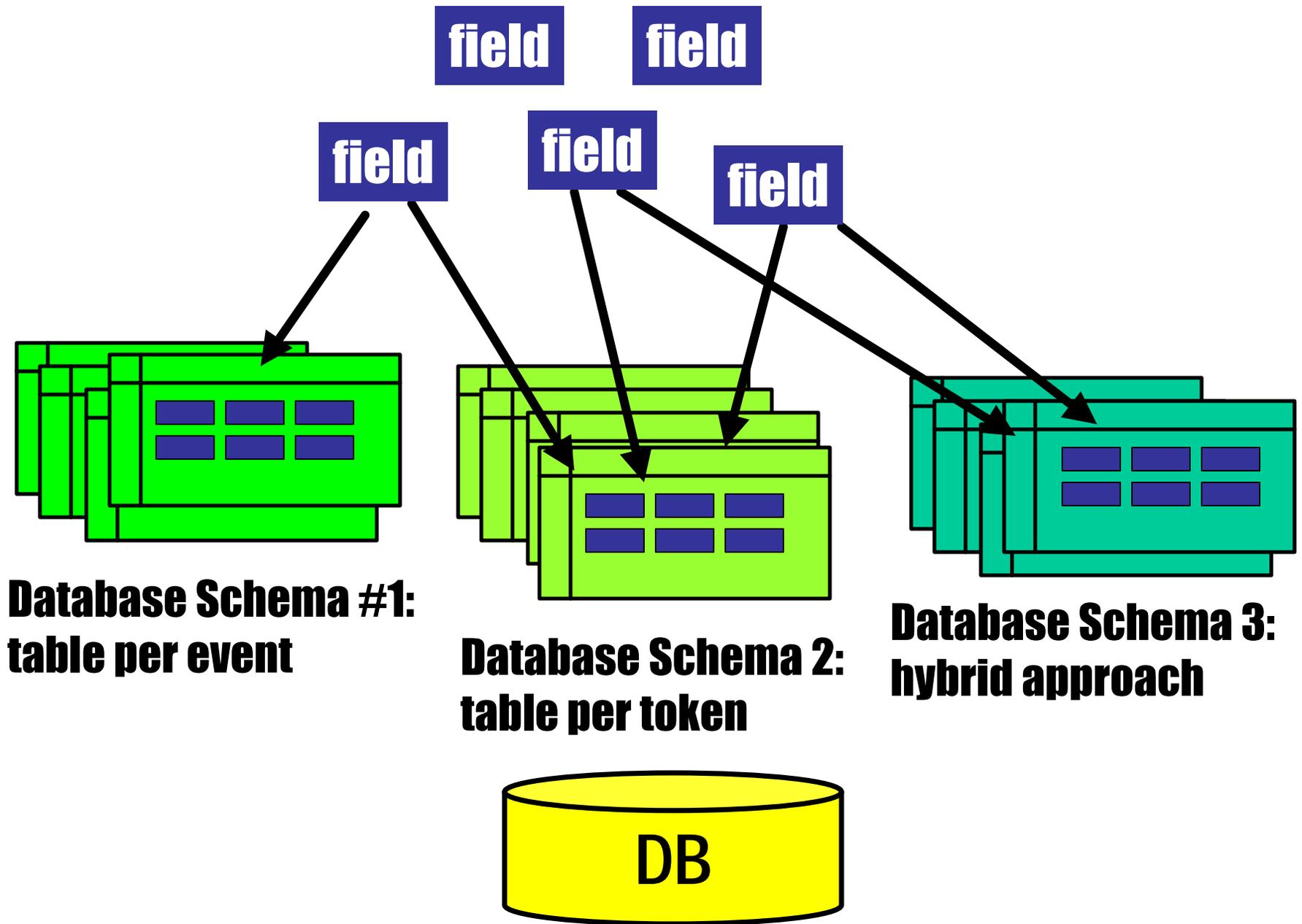
**Audit log file**

**Token**



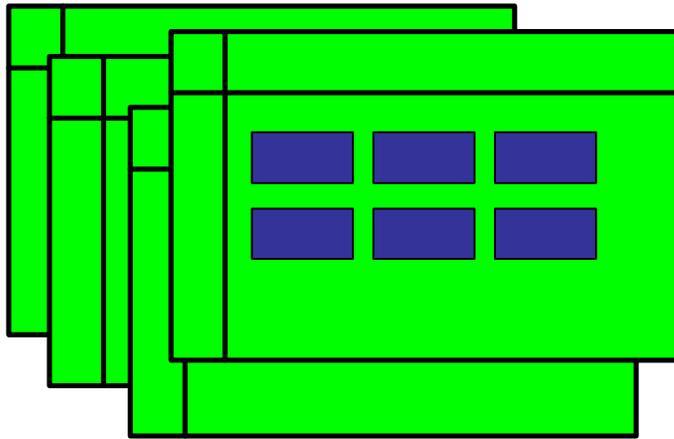
# Decomposing audit records into fields



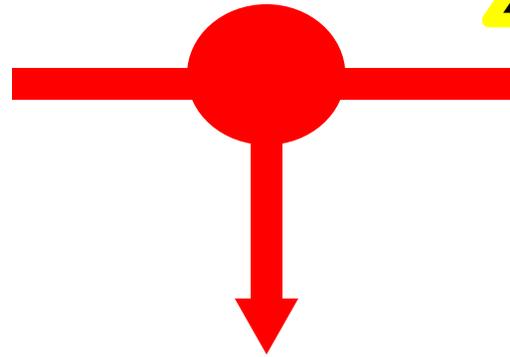
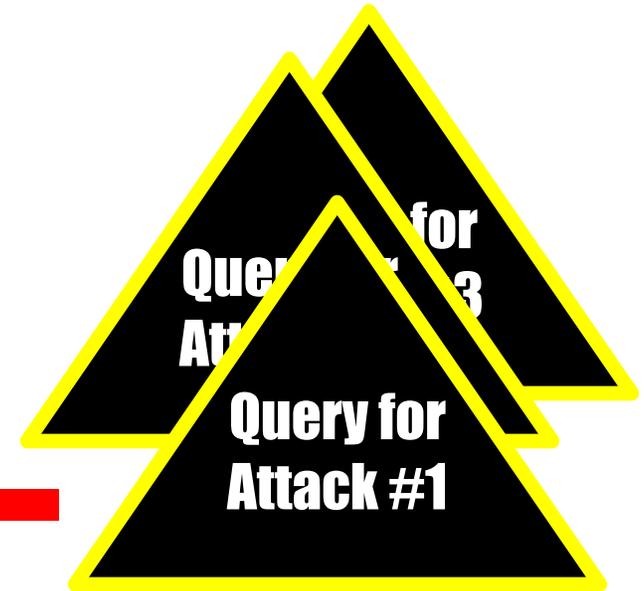


---

## Database tables containing audit data



## SQL detection queries



## Detection report :

**Pattern of Attack #1 detected: User C, Date, Time**

**Pattern of Attack #2 detected: User C, Date, Time**

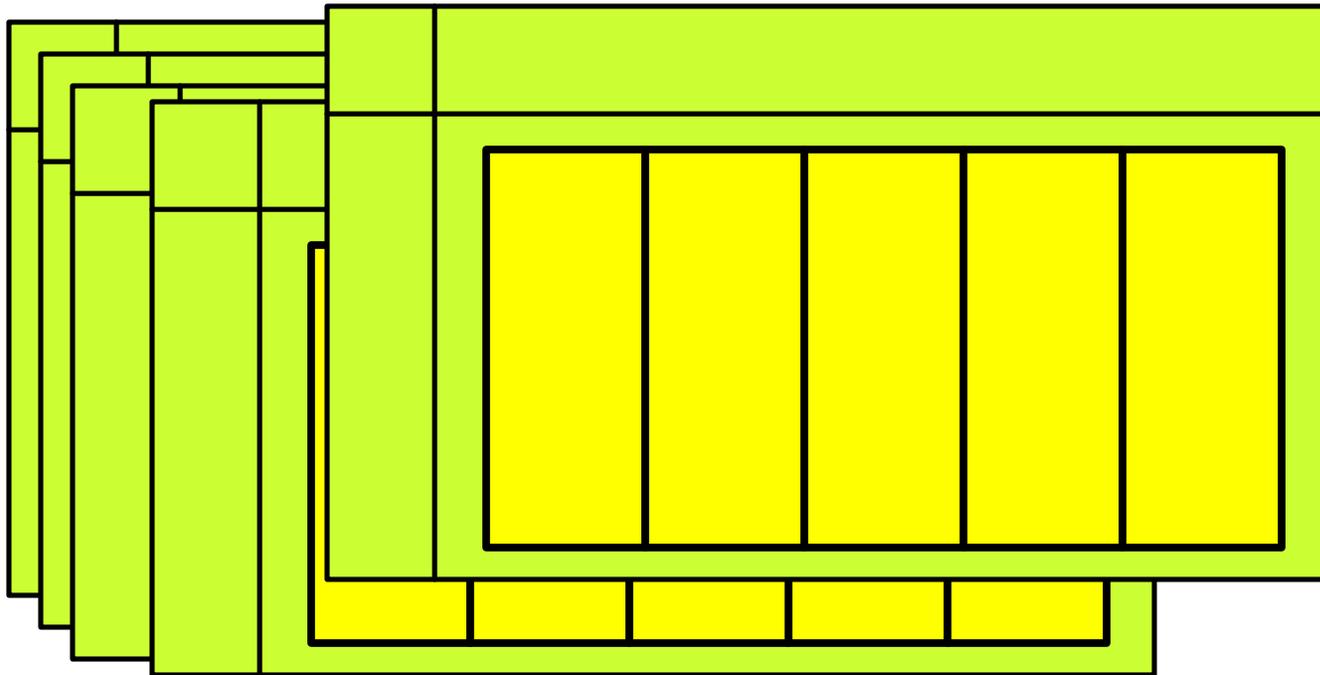
---

---

## **Database Schema 1. Table per event**

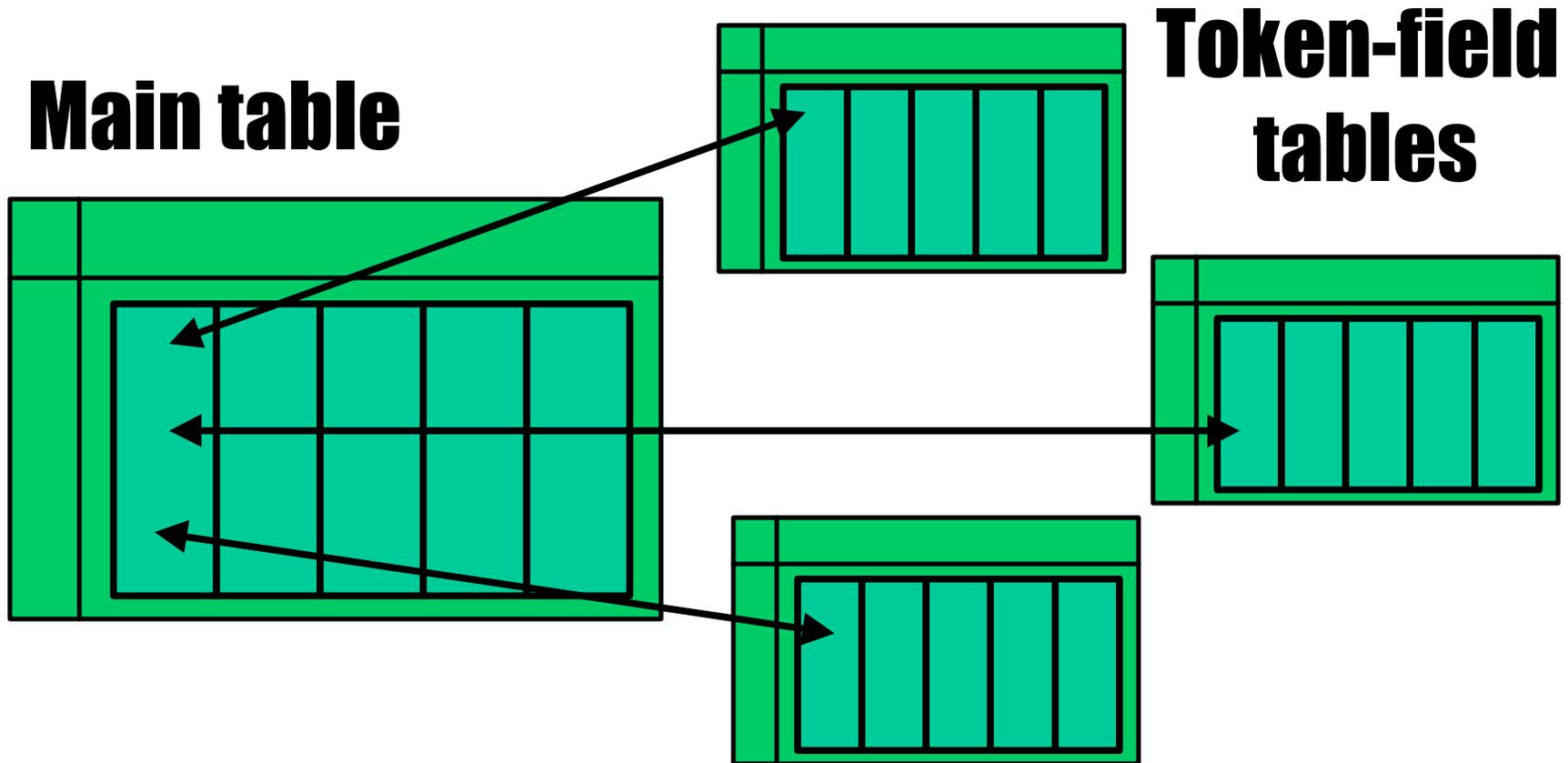
- **Natural implementation from the BSM format**

### **Event tables**



## Database Schema 2. Table per token

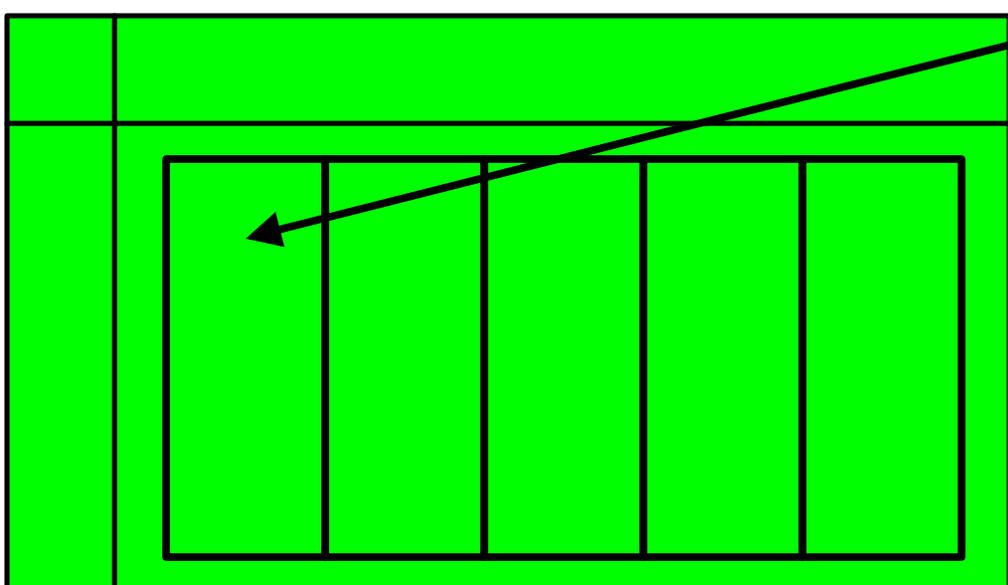
- **Ease to include new events**



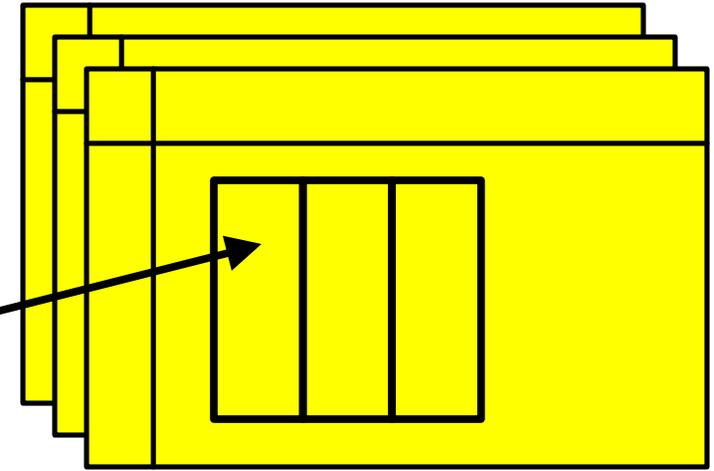
# Database Schema 3. Hybrid approach

- **Remove redundancy and improve efficiency**

**Event table**



A diagram of an Event table. It is a large green rectangle with a black border. Inside, there is a smaller green rectangle with a black border, representing a token field. This token field is divided into five vertical columns by black lines. An arrow points from the top-left corner of this token field to the top-left corner of a token field in a stack of tables on the right.



**Token-field tables**

---

## **Status**

- **3 alternative schemas developed**
- **Scanning, parsing, uploading and detection integrated in one program**
- **Writing more elaborate SQL queries**

## **Points to address**

- **More extensive scalability testing**
  - **More complex attack patterns**
-

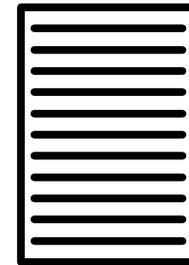
---

# Tools used

**1. Oracle 8. Relational Database Management System**



**2. BSM. Basic Security Module, Sun Microsystems**



**3. ANTLR 2.0.7. Language generator**

**- BSM grammar (Chapman Flack, CERIAS)**



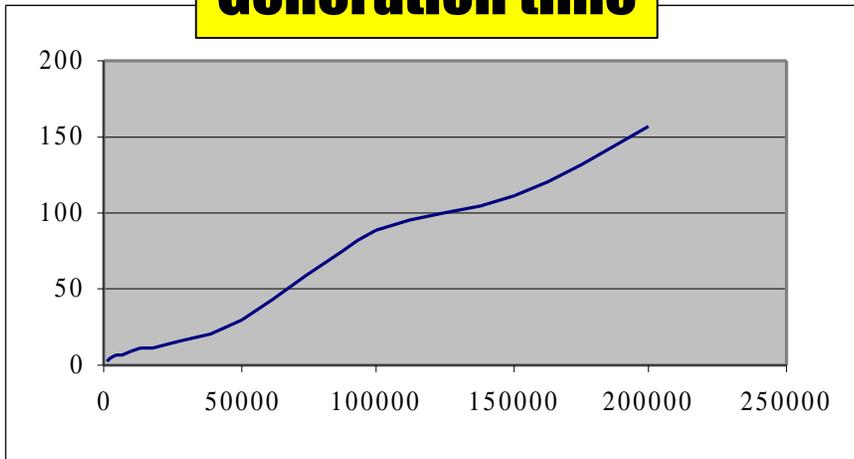
**4. Java 1.2.2**

**5. SunOS 5.6. UNIX scripting (csh)**

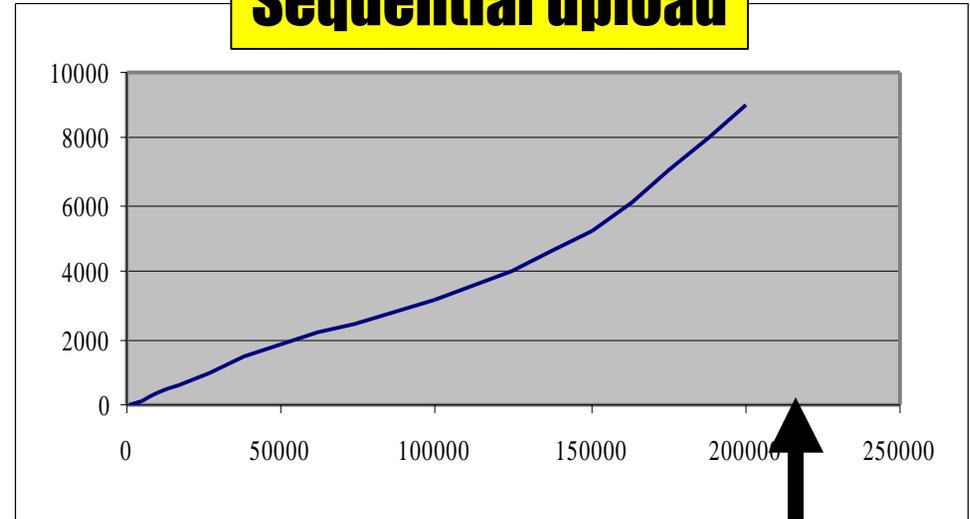
---

# Performance graphs

## Generation time



## Sequential upload



## Detection time



## Batch upload !

