

Purge Patrol



2086 GDPR fines, €4.48 billion
(March 2024)





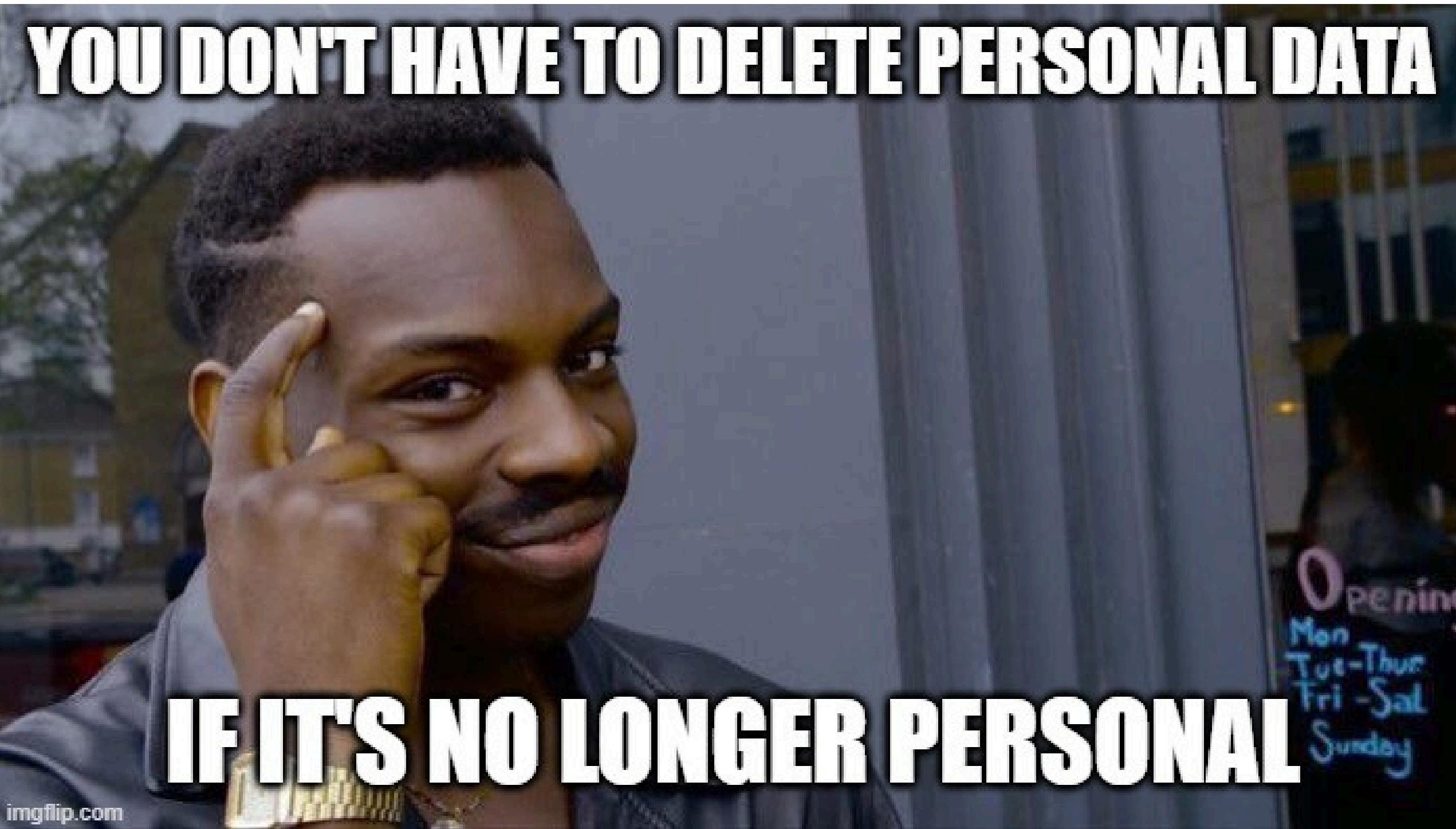
**BUSINESSES MUST
DELETE PERSONAL DATA**



BUT

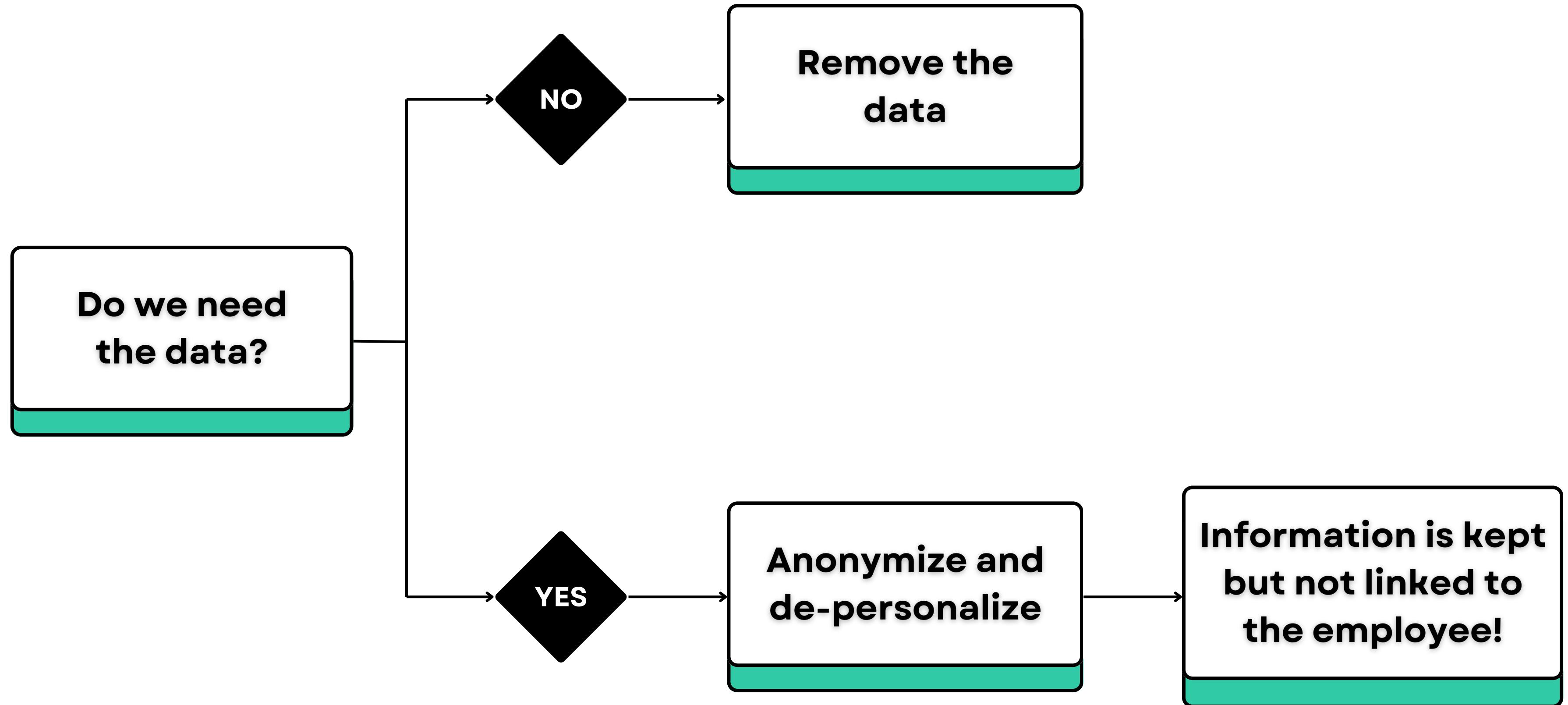
**THIS DATA MAY CONTAIN
INFORMATION WE MUST KEEP**

Our Approach: de-personalizing data



Prototype:

Dealing with employee and project data



Employees database

id	Name	PayZone	Title
1141	Paula Small	Zone C	Senior Manager
...			

Admin database

id	Name	PayZone	Title	Hashing Words
1141	Paula Small	Zone C	Senior Manager	GX7L,nGIY
...				

Project1 database

employee_id	total_hours	start	finish
1141	210	07-23	02-24



Project1 database

employee_id	total_hours	start	finish
de7d5a0e	210	07-23	02-24

Project2 database

employee_id	total_hours	start	finish
1141	290	03-24	10-24



Project2 database

employee_id	total_hours	start	finish
e81fb9c0	290	03-24	10-24



Delete the risk
Keep the rest



TECHNICAL EXAMPLE: REMOVING EX-EMPLOYEES FROM THE DATABASE

```
# Function to remove the employee and anonymize the employee ID occurrences

def remove_employee(employee_id: int):

    # Connect to the specified database
    connection = connect_db()
    cursor = connection.cursor()

    # Step 1: Find all occurrences of the employee ID in the 'Employee_id' column
    cursor.execute(
        "SELECT COUNT(*) FROM employee_worktimes WHERE Employee_id = ?", (employee_id,))
    count = cursor.fetchone()[0]

    # If the employee ID does not exist, exit the function
    if count == 0:
        print(f"Employee ID {employee_id} not found in the database.")
        connection.close()
        return

    # Step 2: Generate a list of random secret words, one for each occurrence
    secret_words = [generate_secret_word() for _ in range(count)]

    # Step 3: Generate distinct hashes using the random secret words
    hashes = [generate_hash(employee_id, secret_word) for secret_word in secret_words]

    # Step 4: Update the 'Employee_id' column with the generated hashes
    cursor.execute(
        "SELECT rowid FROM employee_worktimes WHERE Employee_id = ?", (employee_id,))
    rows = cursor.fetchall()
```

**SMEs constitute 99.3% of the
approximately 3.1 million
enterprises in Germany**

SMALL AND MEDIUM-SIZED ENTERPRISES (SME)

