Qapla: Policy compliance for database-backed systems

Aastha Mehta, Eslam Elnikety, Katura Harvey,
Deepak Garg, Peter Druschel

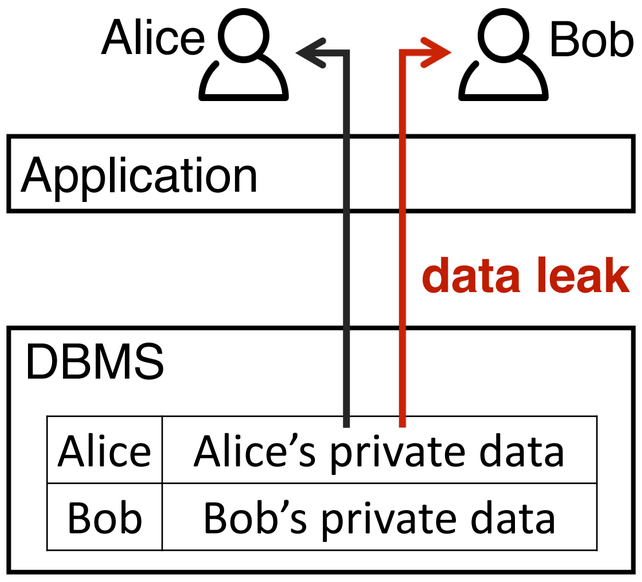
MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

UNIVERSITY OF MARYLAND 18 56

1. Applications with **confidential data** need to prevent **data leaks**

Confidential data may include:

- **Patient records** in healthcare systems
- **Customer purchase data** on e-commerce websites
- **Salaries** and **ages** in personnel management systems



Access policies: what data a user is authorized to access

Data leak: a user gets access to unauthorized data

Data leaks cause:

- Loss of user privacy
- Loss of revenue and reputation for providers

2. Current approach: enforce policies **in code**

```
def index(request):
    user = request.user
    roles = map(_role_to_name, Role.objects.filter(user__exact=user))
    if ('Faculty' in roles) or ('DoctoratesOnly' in roles)
      or ('InternsOnly' in roles) or ('FacultyOnly' in roles)
      or ('Office' in roles):
        if ('Faculty' in roles) or ('Office' in roles):
          context['show_status'] = True
        app_type = request.GET.get('app_type', 'faculty')
        if not allowed_to_see_app_type(app_type, roles):
          if 'DoctoratesOnly' in roles:
              app_type='doctorate'
          if 'InternsOnly' in roles:
              app_type='intern'
          if 'FacultyOnly' in roles:
              app_type='faculty'
        # ...Application logic...
        if 'Office' not in roles:
          applications = applications.filter(created_date__gt=user.date_joined)
        # ...Application logic...
        return render_to_response(app_template, context}

def allowed_to_see_app_type(app_type, roles):
  if ('Faculty' in roles) or ('Office' in roles):
    return True
  if (app_type=='doctorate') and ('DoctoratesOnly' in roles):
    return True
  if (app_type=='intern') and ('InternsOnly' in roles):
    return True
  if (app_type=='faculty') and ('FacultyOnly' in roles):
    return True
  return False
```

policy

policy

app logic

policy

app logic

policy function

What is the problem?

Bugs in code

↓

Policy violations

↓

Data leaks

Goal:
Separate policy **specification** & **enforcement** from application code

3. Qapla policy specification

Qapla policies are:

- On DB schema – separate from application
- Easy to express – specified in SQL
- Rich – constraints on DB values, joins, aggregates

Example: Personnel management system

Employees

empID	dept	name	age
111	HR	Alice	24
222	Finance	Bob	49

Linking two columns can reveal more information than reading them separately

Policy:
All employees can read names and ages individually, but only the Human Resources (HR) department may link them together, and an employee may link her own fields

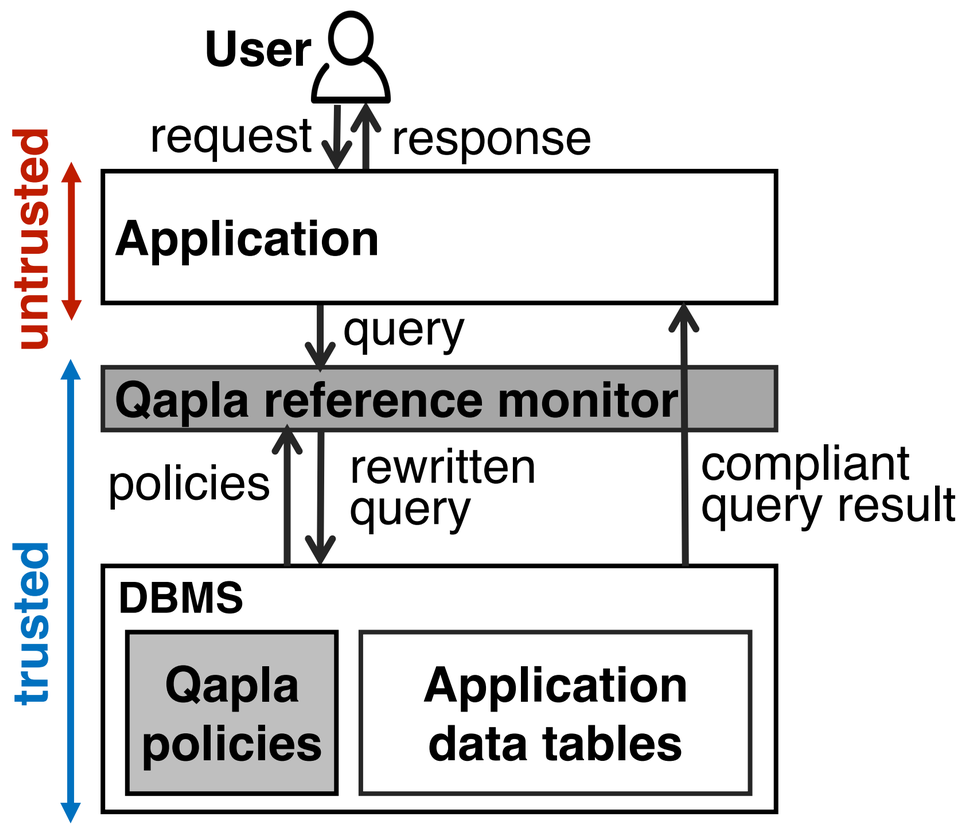
Specification:

{name, age} :-
`Employees:((empID = $user) OR`
`EXISTS(SELECT 1 FROM Employees`
`WHERE empID = $user AND dept = HR))`

4. Qapla policy enforcement

Qapla reference monitor:

- Intercepts application accesses to DB
- Rewrites queries to comply with applicable policies



Performance:
Overheads are moderate and do not noticeably impact end-user latency