gedafe

THE GENERIC DATABASE FRONTEND

Gedafe: a Generic Web Front-end for PostgreSQL

David Schweikert, System Administrator
IT Support Group, Departement of Electrical Engineering
Swiss Federal Institute of Technology, Zurich
dws@ee.ethz.ch

O'Reilly Open Source Convention 2001
Thursday, July 26 4:30pm - 5:15pm, Bel Aire South in the West Tower

Outline

- Goal: A Time-Tracking Application

- Usual Approach to the Problem

- Application in the Database Approach

- Gedafe Implementation

- Gedafe Features

- Future Work

- Questions and Feedback

Goal: A Time-Tracking Application

- Web-based workgroup time-tracking

- "People", "Time" and "Projects" (PTP)

- Reason for developing Gedafe

- Example application in this talk

Netscape: ISG People Time and Projects

File   Edit   View   Go   Communicator   Help

ISG People Time and Projects

User: **dws**   Entry | Refresh | Logout | Docs

A Absences   A Work/Cost Log   B Customers   B Project Budget   B Project Price Override   B Project Scheduler   B Projects   C Bills sent
Z Absence Types   Z Bill Item Types   Z Genders   Z Personnel   Z Project Schedule Entry Type   Z Units   Z Work/Cost Types

A Work/Cost Log

Search: Date ▭   2000-10-23   Search   Show All

ID	Date	Worker	Project	W/CT	Amount	Vrt	Description		
5039	2000-10-23	dws	ISG: Training/Education	WORK	0.50 hrs	no	slashdot, etc.	✎	⚡
5057	2000-10-23	dws	ISG: Windows 2000 Konzept	WORK	3.50 hrs	no	work on ISG::Util an [...]	✎	⚡
5038	2000-10-23	dws	ISG: People Time and Projects	WORK	0.50 hrs	no	taken ANNOUNCE chang [...]	✎	⚡
5056	2000-10-23	dws	LKT: Support	WORK	0.33 hrs	no	RT 1526: password on [...]	✎	⚡
5054	2000-10-23	dws	ISG_: Personal	WORK	0.33 hrs	no	work log	✎	⚡
5052	2000-10-23	dws	IFH: Unix System Reorganization	WORK	0.42 hrs	no	fixed tecplot-8.0-mo	✎	⚡
5051	2000-10-23	dws	EE: Unix Support Students	WORK	1.50 hrs	no	problems with accoun [...]	✎	⚡
5050	2000-10-23	dws	EE: Unix Support Students	WORK	0.17 hrs	no	RT 1527: java servle [...]	✎	⚡
5049	2000-10-23	dws	EE: NT Terminal Server	WORK	0.67 hrs	no	problems with zhadum [...]	✎	⚡
5037	2000-10-23	dws	ISG_: eMail	WORK	0.33 hrs	no	daily e-mail	✎	⚡

ADD   Next -->

ISGPTP the *ISG Time Tracker Db*
Db Design and Programming by Tobias Oetiker and David Schweikert
IT Support Group @ D-ELEK / ETH Zürich

MADE WITH gedafe   PostgreSQL

100%

Netscape: ISG People Time and Projects

File   Edit   View   Go   Communicator   Help

ISG People Time and Projects   User: **dws**   Entry | Refresh | Logout | Docs

A Absences   A Work/Cost Log   B Customers   B Project Budget   B Project Price Override   B Project Scheduler   B Projects   C Bills sent
Z Absence Types   Z Bill Item Types   Z Genders   Z Personnel   Z Project Schedule Entry Type   Z Units   Z Work/Cost Types

Edit Work/Cost Log

Date:	2000-10-19
Worker:	[ ] dws -- Schweikert, David
Project:	[ ] 19 -- ISG: People Time and Projects, dws
W/C Type:	[ ] COMM -- Communication (phone, email, talk, moo) / hrs
Amount:	0.75 Virtual: ☐
Description:	discussion with tobi about name change GPW3F -> Gedafe, PearlReports integration, web-site, slides
Bill:	

Update Reset Cancel

ISGPTP the *ISG Time Tracker Db*
Db Design and Programming by Tobias Oetiker and David Schweikert
IT Support Group @ D-ELEK / ETH Zürich

MADE WITH gedafe   PostgreSQL

Netscape: ISG People Time and Projects

File   Edit   View   Go   Communicator   Help

ISG People Time and Projects

User: **dws**

Entry | Refresh | Logout | Docs

A Absences   A Work/Cost Log   B Customers   B Project Budget   B Project Price Override   B Project Scheduler   B Projects   C Bills sent
Z Absence Types   Z Bill Item Types   Z Employers   Z Genders   Z Personnel   Z Project Schedule Entry Type   Z Units   Z Work/Cost Types   Z Workareas

Database Error

The database backend produced the following error:

ERROR: The worklog accepts only records for the current users

Back to the Form!

ISGPTP the *ISG Time Tracker Db*
Db Design and Programming by Tobias Oetiker and David Schweikert
IT Support Group @ D–ELEK / ETH Zürich

MADE WITH
gedafe

PostgreSQL

Usual Approach to the Problem: LAMP

- Dumb database (no logic, just tables)

- PHP or Perl scripts implement the application logic:

  - Authentication and authorization
  - Presentation of data
  - Data validation
  - Referential integrity

LAMP Architecture
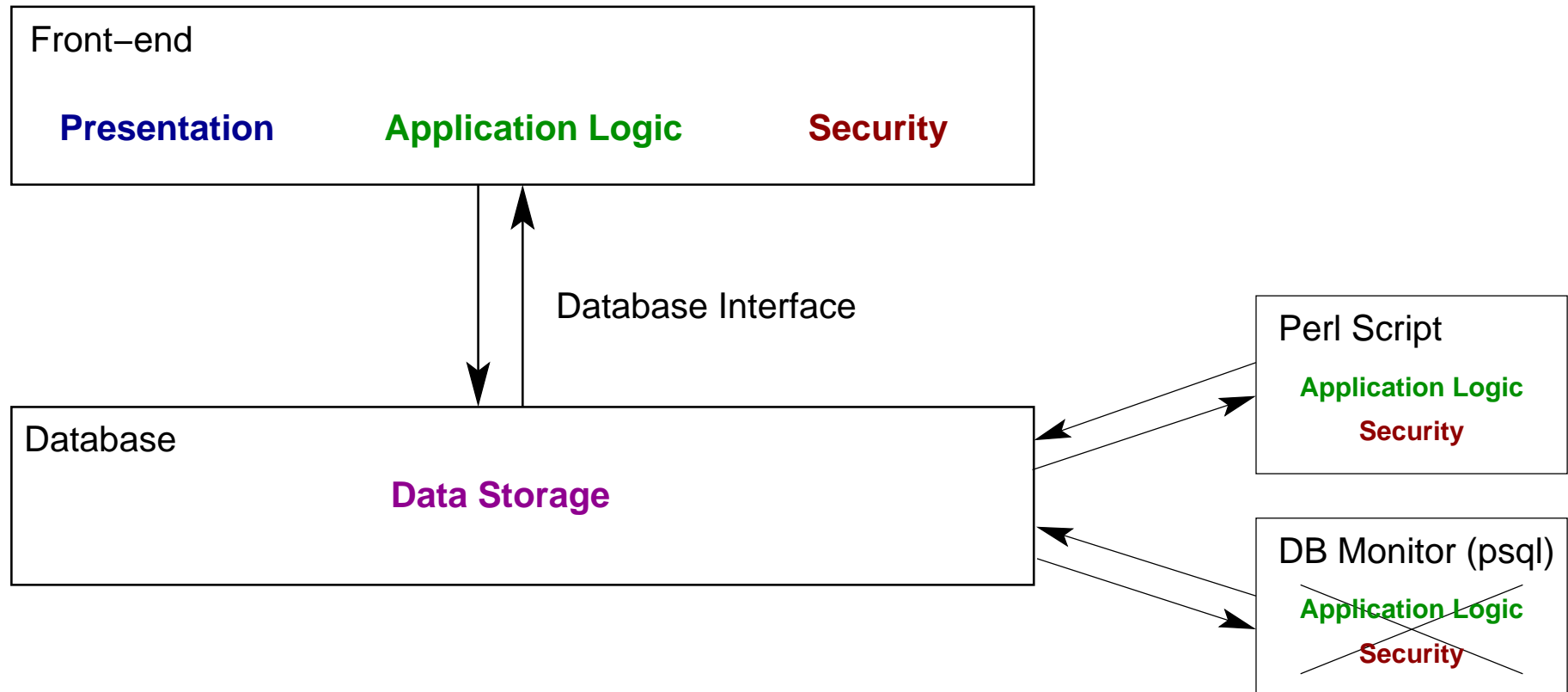
LAMP Problems

- Front-end code grows with the application complexity

- Code embedded in HTML difficult to read

- Potentially inconsistent or incomplete user interface

- Security and data integrity are not enforced if another front-end is used

Application in the Database Approach

- In the database:

  - Security (authentication and authorization)
  - Information on how to present the data
  - Application logic

- Possible with databases that support:

  - Referential integrity constraints
  - Presentation views
  - Data validation through rules
  - Triggers
  - Stored procedures

- Lightweight, generic front-end just presents the data

Gedafe Architecture
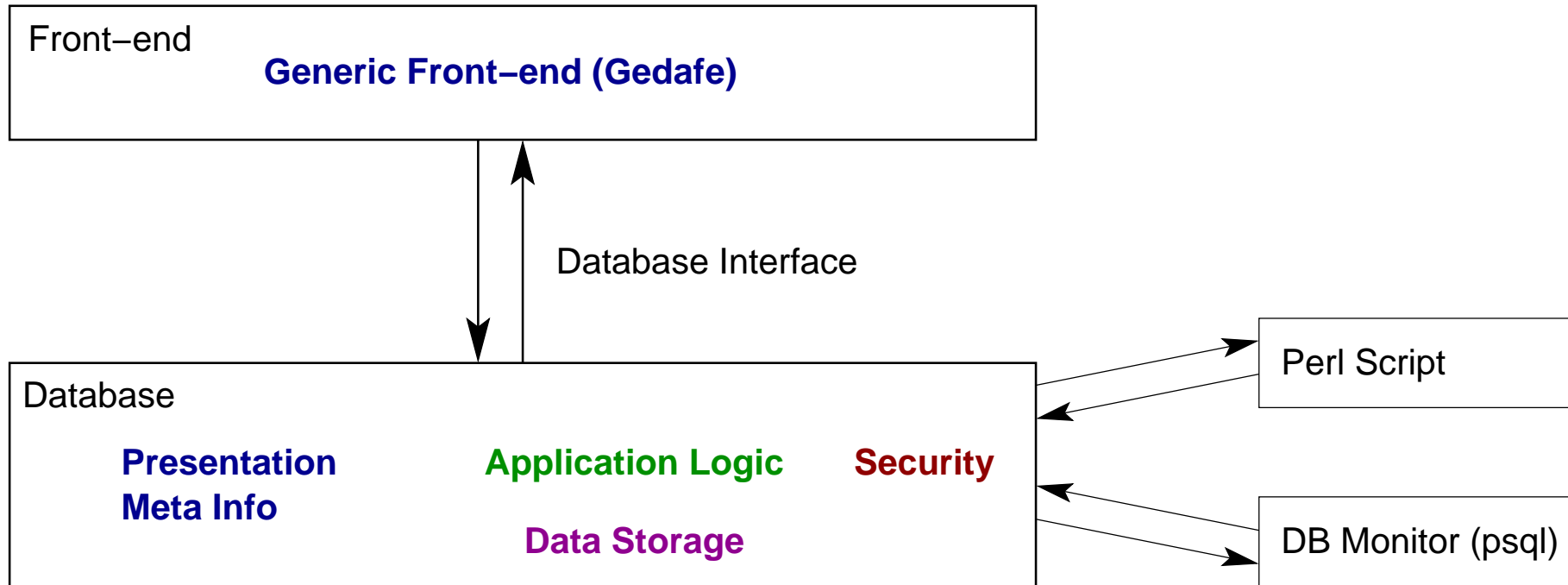
Advantages of the Application in the Database Approach

- Security and data integrity enforced (front-end independent)

- Less code (database only)

- Triggers simplify implementation of application logic

- Write once the front-end and reuse it for all applications

Gedafe - the Generic Database Front-end

- Generic front-end for the "Application in the Database" approach

- All tables full-featured and consistent

- Flexible HTML templates

- "Double form submission protection" and "deep linking"

- Simple to install, proven to work

- Perl-only, portable

Application Script Example

```
#!/usr/bin/speedy -w

use lib '/usr/local/gedafe/lib/perl';

use Gedafe::Start;

Start(
    db_datasource  => 'dbi:Pg:dbname=demo',
    templates      => '/usr/local/gedafe/demo/templates',
);
```

- Gedafe is a library

- Application script takes 6 lines of Perl code

Gathering of Database Information

- Gets all information about the application from the database at login

- Database information is cached

- No performance penalty with persistent Perl (SpeedyCGI or mod_perl)

Internals: Tables and Fields

```
CREATE TABLE product (
    product_id              SERIAL  NOT NULL PRIMARY KEY,
    product_hid             CHAR(5) NOT NULL UNIQUE,
    product_desc    TEXT     CHECK (product_desc != '')
);
```

- Field type determines editing widget

- Key must be called table_*id* (never changes)

- "Human-speakable ID": table_*hid* (can be changed)

Internals: Descriptive Names

```
COMMENT ON TABLE product IS 'Products';
COMMENT ON COLUMN product.product_hid IS 'Part Nr.';
COMMENT ON COLUMN product.product_desc IS 'Description';
```

- COMMENTs are the "nice name"

Internals: References

```
CREATE TABLE orders (
    orders_id        SERIAL NOT NULL PRIMARY KEY,
    orders_date      DATE   NOT NULL DEFAULT CURRENT_DATE,
    orders_customer  INT4   REFERENCES customer,
    orders_product   INT4   REFERENCES product,
    orders_qty       INT4,
    orders_shipped   BOOLEAN
);
```

- Used to edit orders_customer and orders_product

- Expects *customer_combo* and *product_combo* views

Internals: Combo Boxes

```
CREATE VIEW product_combo AS
        SELECT product_id AS id,
                product_hid || ' -- ' || product_desc AS text
        FROM product;
```

- id column is key for a record of referenced table (product)

- text column is description of that record for the combo-box

Internals: Presentation Views

```
CREATE VIEW orders_list AS
   SELECT orders_id, orders_date, customer_name, orders_qty,
          product_hid, product_desc, orders_shipped
   FROM   orders, customer, product
   WHERE  customer_id = orders_customer AND
          product_id = orders_product;
```

- View must be called `table_list`, used automatically

- First column is key for the record to edit

Internals: Additional Meta Data

```
INSERT INTO meta_fields VALUES
    ('product', 'product_desc', 'widget', 'area');

INSERT INTO meta_fields VALUES
    ('orders', 'orders_date', 'copy', '1');
```

- Specification of additional presentation specifications

- Information that not available from PostgreSQL internal tables

- (*attribute*, *value*) pairs

- meta_database, meta_tables, meta_fields

Deep Linking

- Every page is bookmarkable

- Use URL-encoding whenever possible (GET requests)

- Transparent authentication

Authentication

- Authentication at the database level

- Cookie → User/Password mapping is stored by a daemon

Double Form Submission Protection

Problem:

- Multiple submission of same form (through page reload)

- Duplicate data could be entered in the database accidentally

Solution:

- Each form has a unique *form-id* that is stored by the Gedafe daemon

- Subsequent submissions with same form-id are rejected

Future Work

- Multi-select, m:n relations

- Editable views

- Naming conventions optional

- Port to other databases

- Improve or change the HTML template system

- Use PearlReports to create complex reports

Thanks

More information on

http://isg.ee.ethz.ch/tools/