Discussion Topic

On the use of Database Technology in Embedded Control Programs

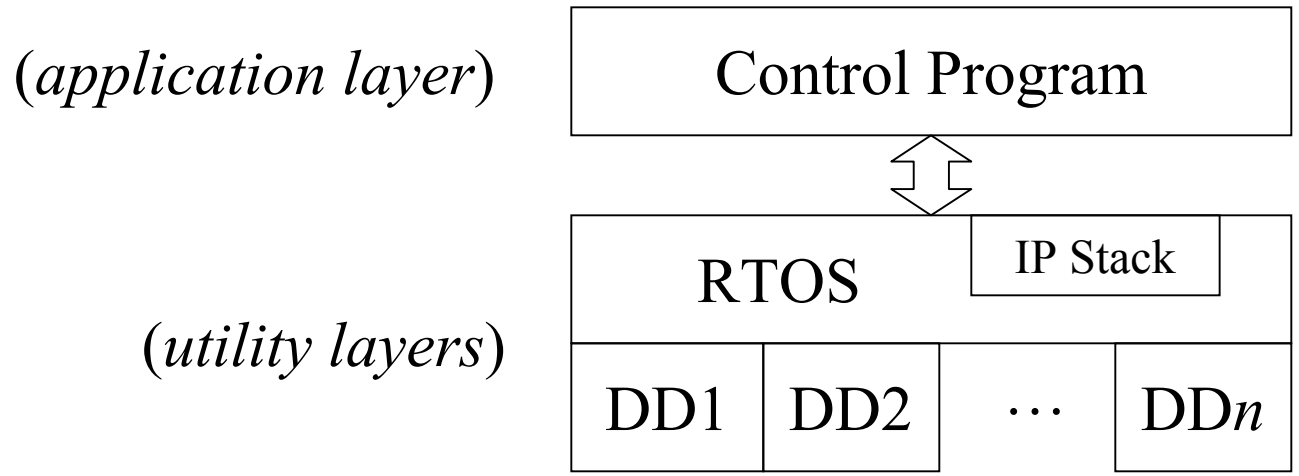
Moderator: Grant Weddell

Embedded Control Program (ECP)

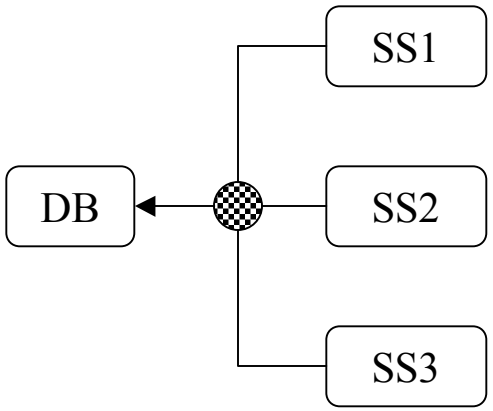
E.g.: *The software part of an intelligent device.*

An Intelligent Device

Software Components



Reference Architecture



Ã (*responsible for real time functionality*)

Ã (*responsible for administration*)

Ã (*responsible for diagnosis*)

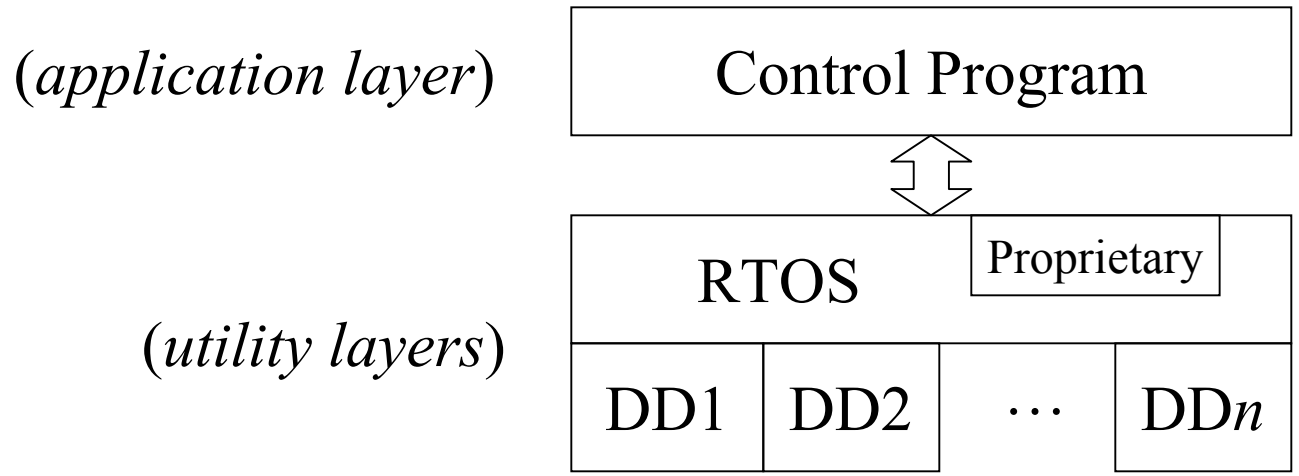
Definition: *Any software system for which there is utility in adopting a repository style of architecture.*

(*unspecified*)

Another Phone

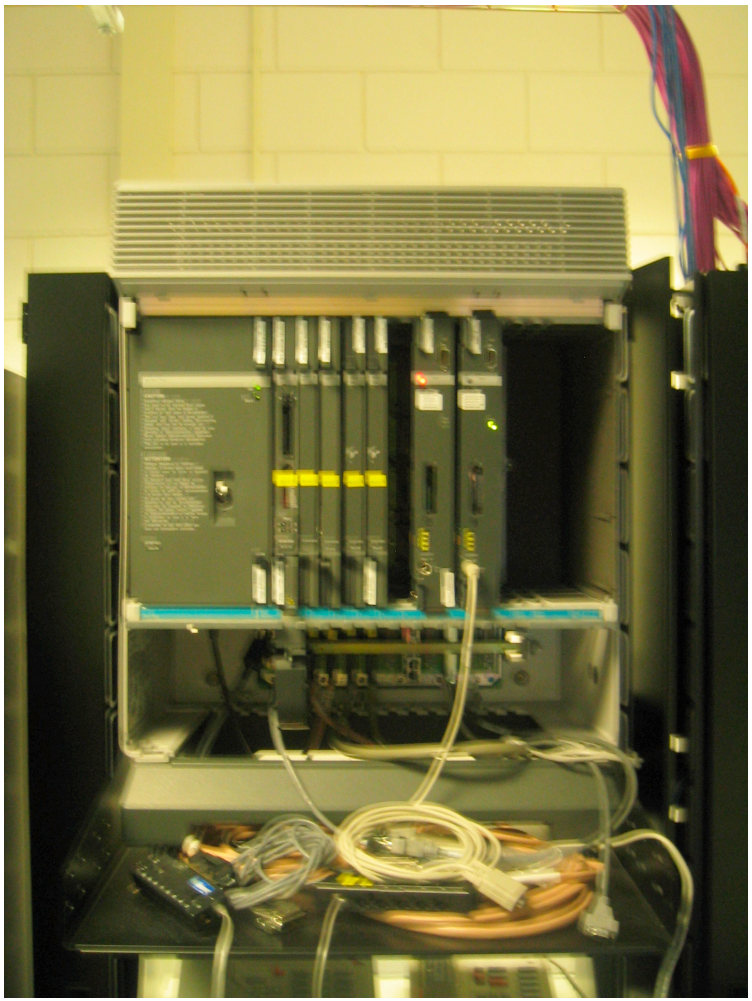
And Another

Software Components

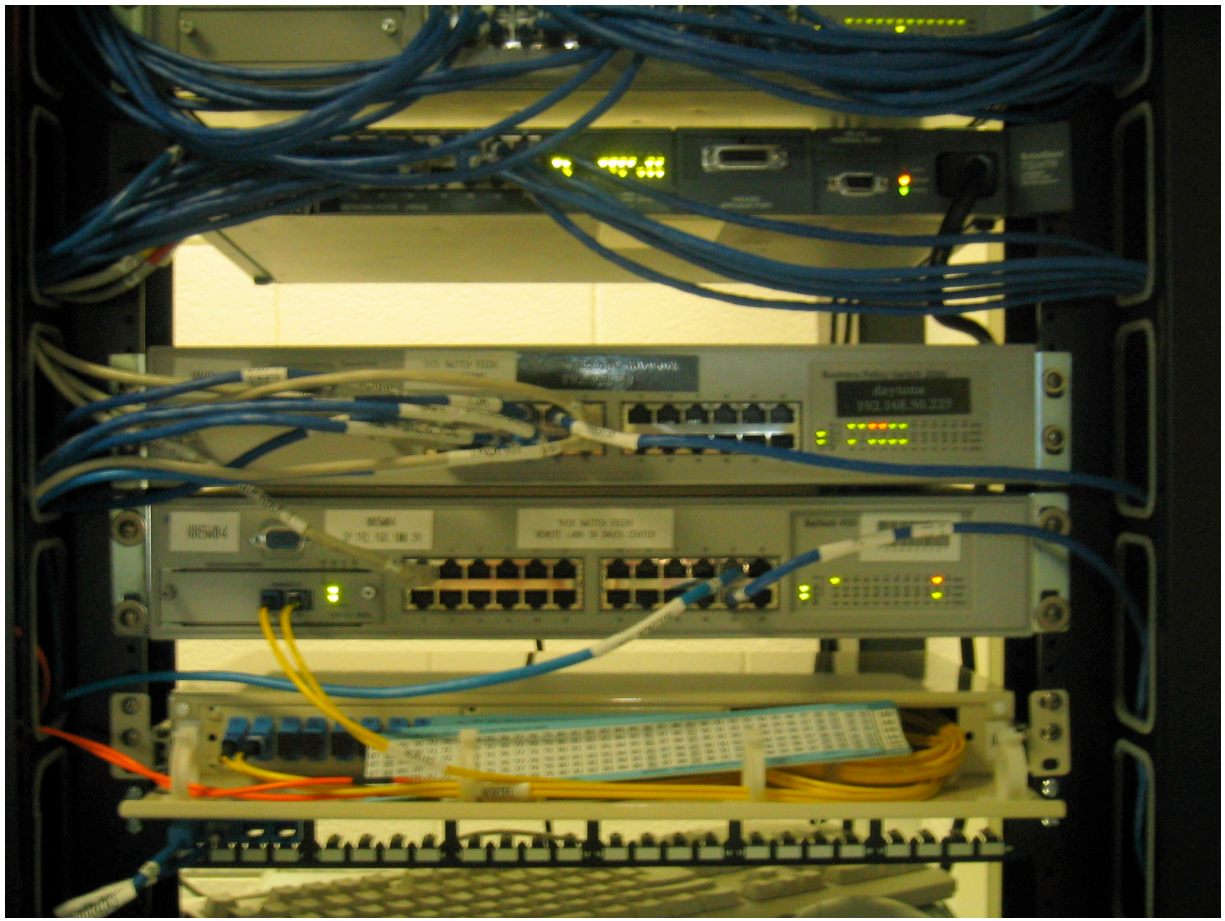


And Another

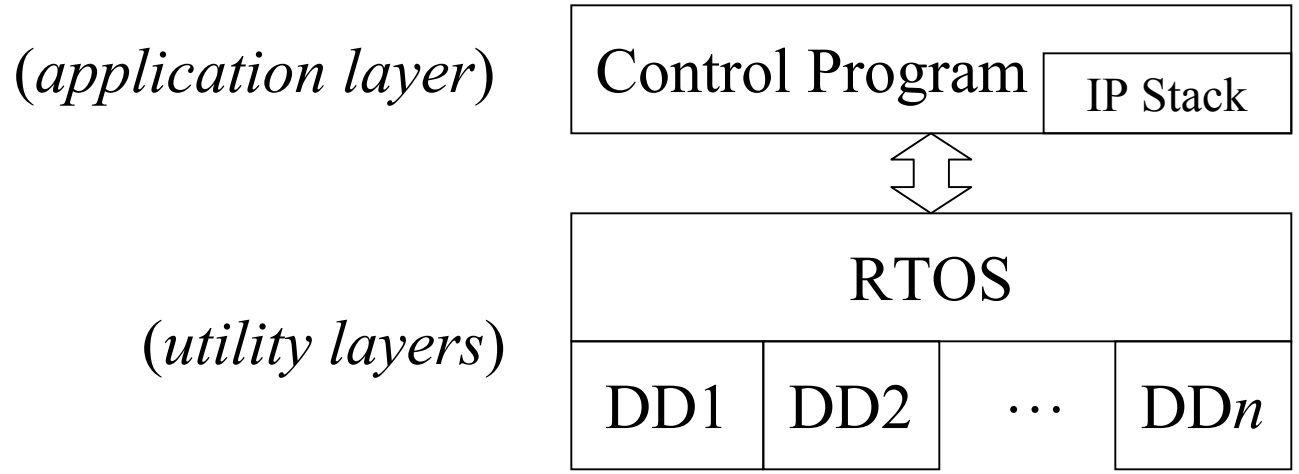
A Telephone Switch



Data Switch



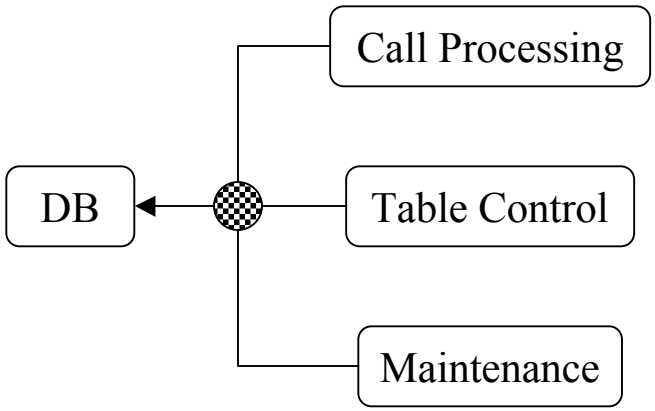
Software Components



Vending Machine

Automated Banking Machine

ECP for a Telephone Switch



Ã (*responsible for real time functionality*)

Ã (*responsible for administration*)

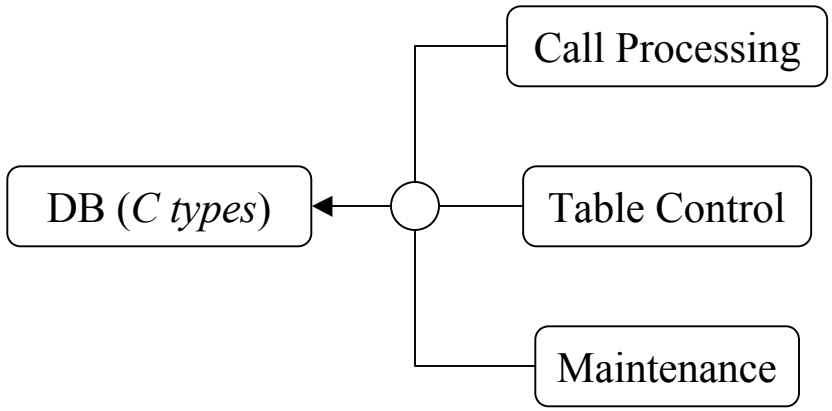
Ã (*responsible for diagnosis*)

Contents of DB

- Information about subscribers.
- Network status.
- Call state data.
- Diagnostic information.

(*unspecified*)

Effect of Performance Requirements



Ã (*responsible for real time functionality*)

Ã (*responsible for administration*)

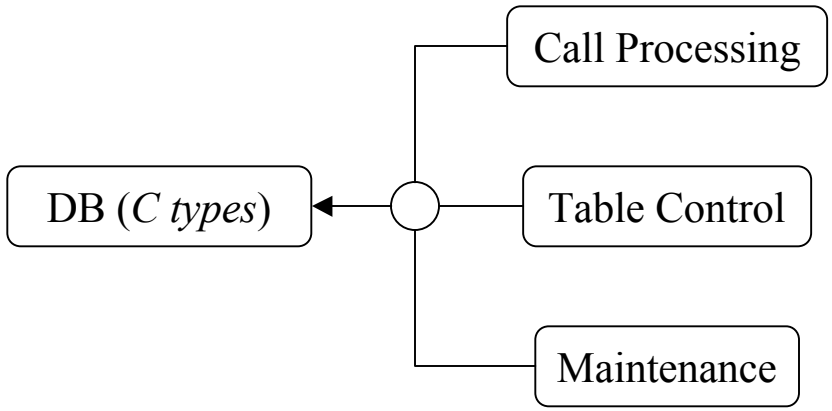
Ã (*responsible for diagnosis*)

Contents of DB

- Information about subscribers.
- Network status.
- Call state data.
- Diagnostic information.

◯ API

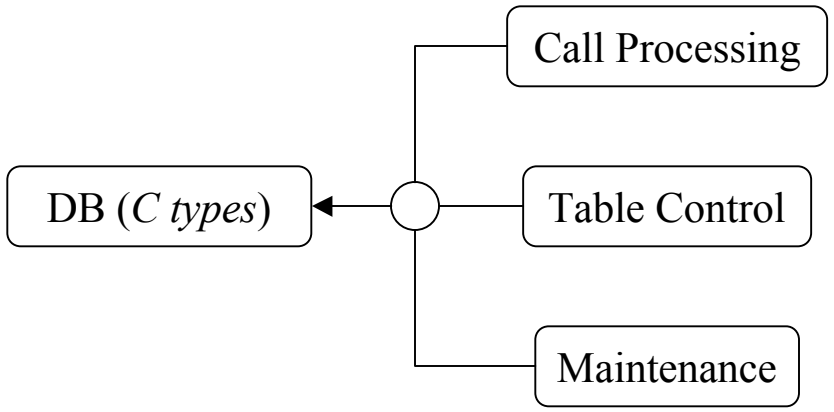
Data Access



- Access a record field.
- Access *i*th entry of an array.

◯ API

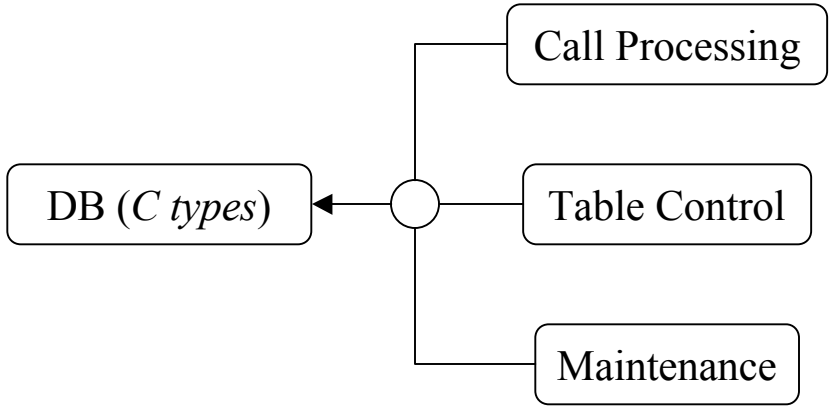
Data Revision



- Update a record field.
- Update *i*th entry of an array.
- Allocate space for a record.
- Free space for a record.

◯ API

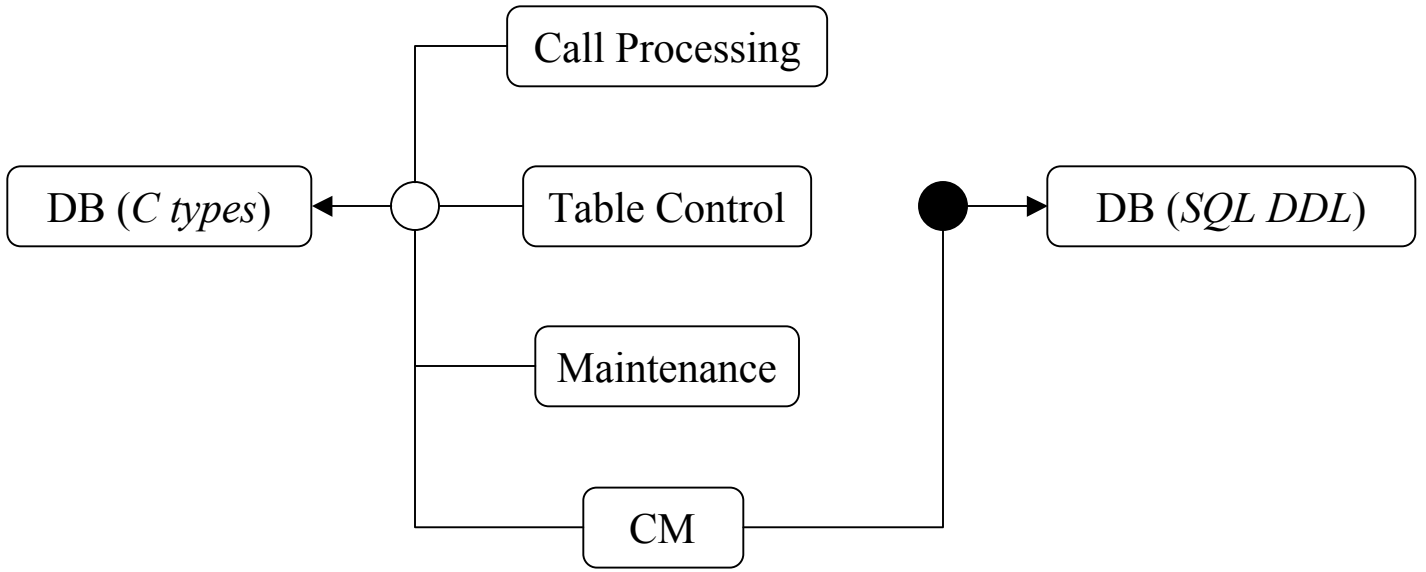
Transaction Management



- Successful test and set (*a record field is zero; set the field to one*).
- Unsuccessful test and set (*a record field is non-zero*).

◯ API

Integrating Telephone Systems



-

(*responsible for cache management*)

○ API

● SQL DML

Data Access

API

- Access a record field.
- Access *i*th entry of an array.

SQL DML

- Access a tuple attribute value.
- Open an iterator
  (*defined by a static SQL query*).
- Succeed in accessing an iterator.
- Fail in accessing an iterator.
- Increment an iterator.
- Close an iterator.

Data Revision

API

- Update a record field.
- Update *i*th entry of an array.
- Allocate space for a record.
- Free space for a record.

SQL DML

- Update a tuple attribute value.
- Create a new tuple.
- Delete an existing tuple.

Transaction Management

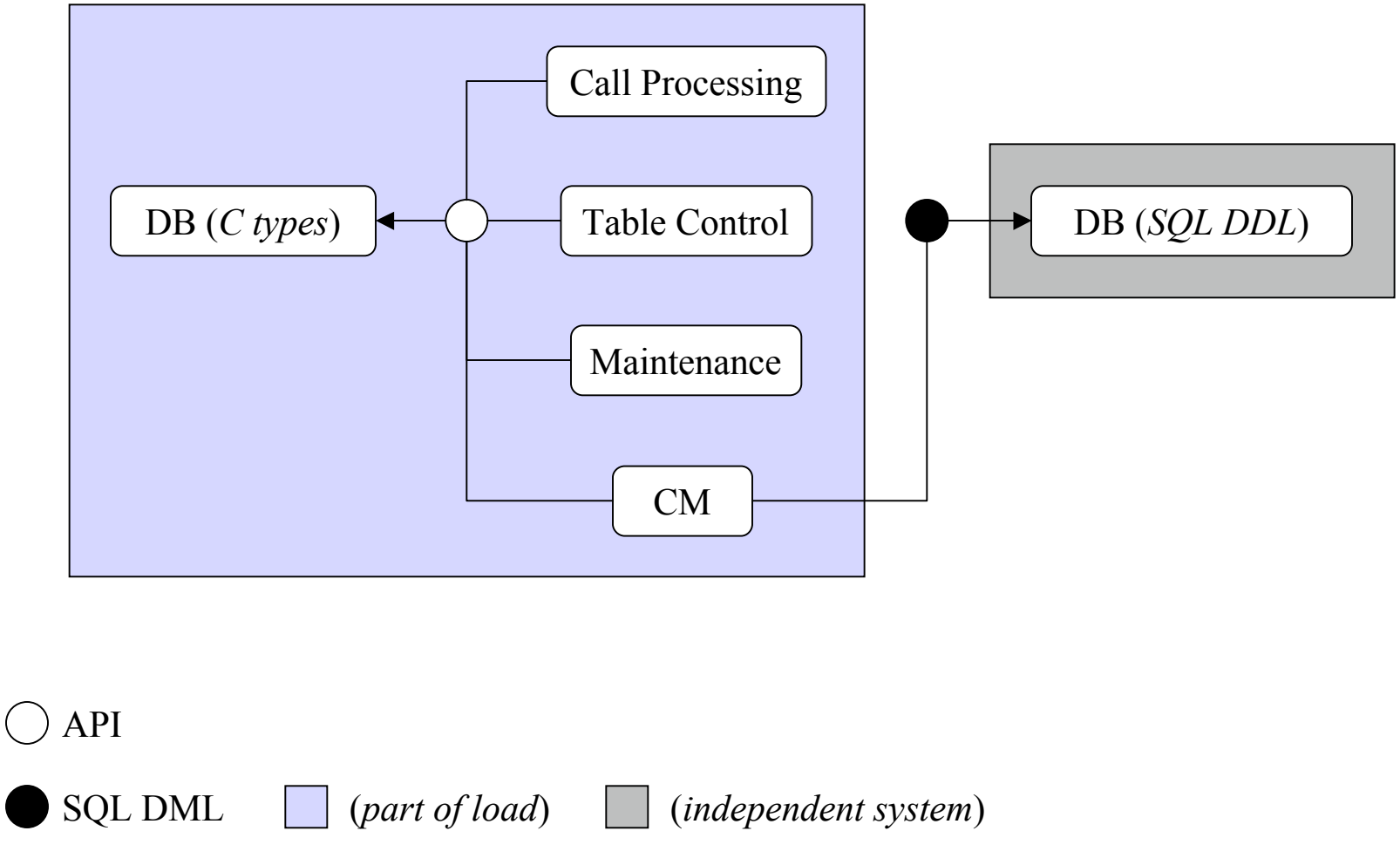
API

- Successful test and set
  (*a record field is zero; set the field to one*).
- Unsuccessful test and set
  (*a record field is non-zero*).

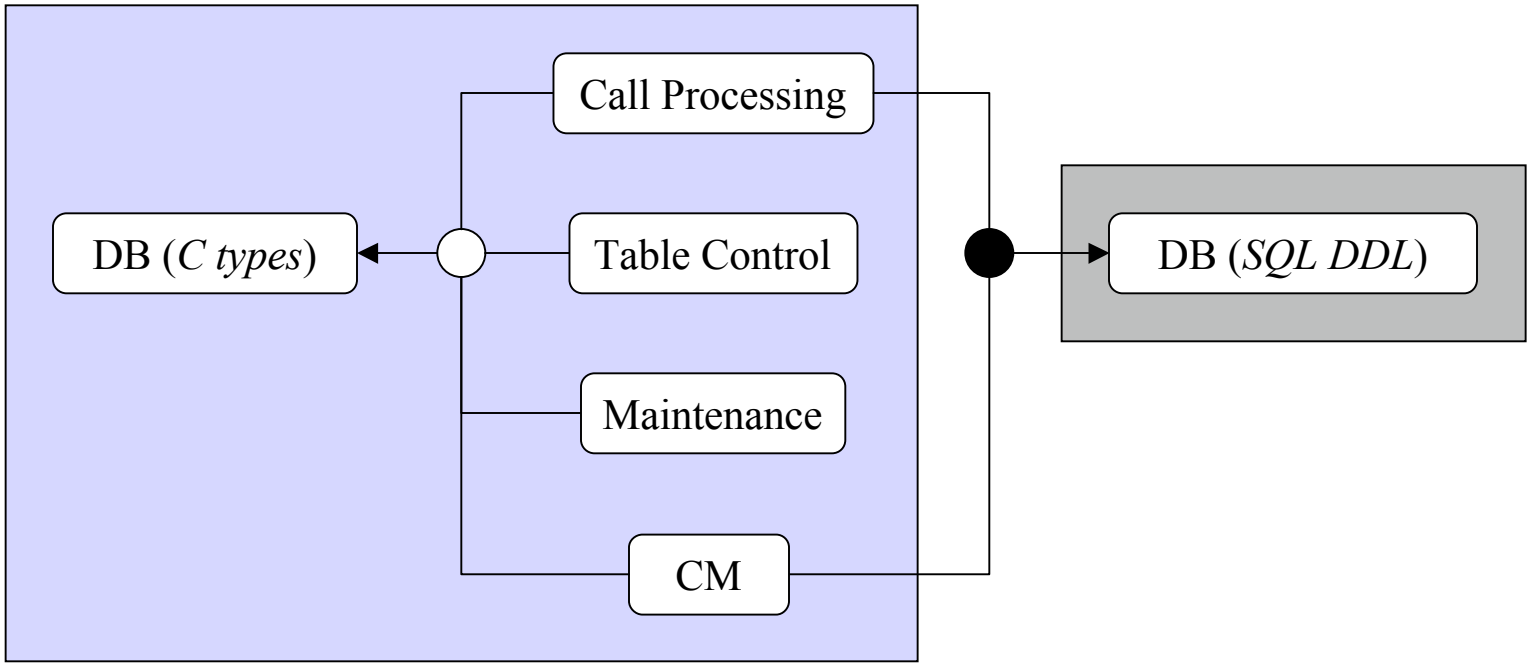
SQL DML

- Connect
- Begin transaction.
- Commit transaction.
- Abort transaction.
- Disconnect

Packaging


Call Processing
DB (*C types*)
Table Control
DB (*SQL DDL*)
Maintenance
CM
API
SQL DML
(*part of load*)
(*independent system*)

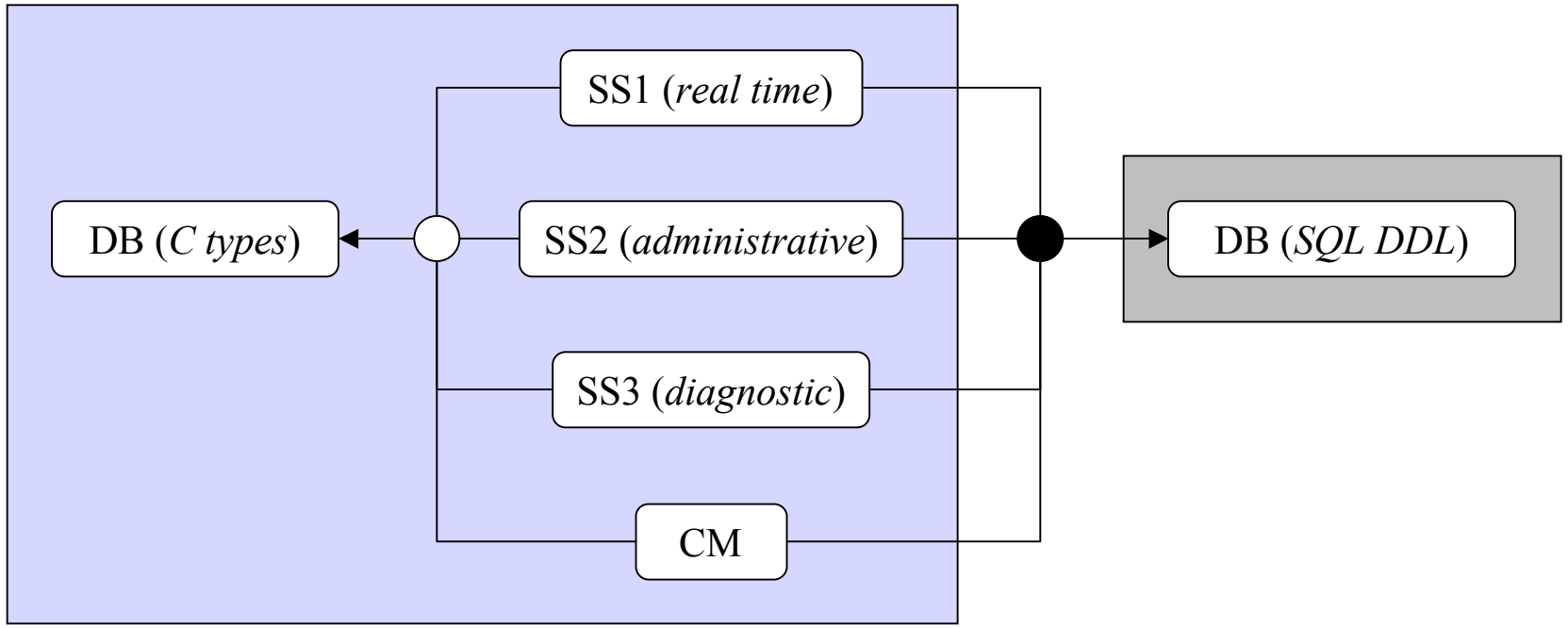
Advanced Intelligent Networks



API

SQL DML (*part of load*) (*independent system*)

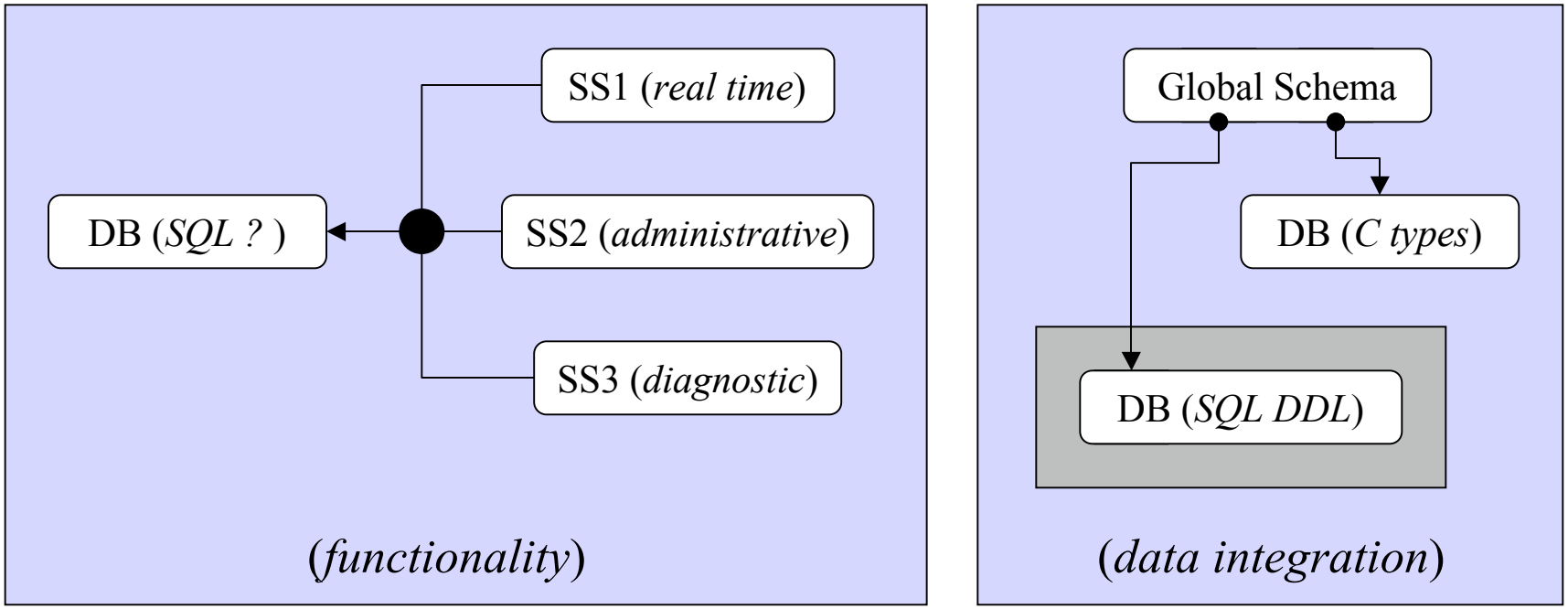
ECP: Generic Runtime Architecture



API

SQL DML (*part of load*) (*independent system*)

ECP: Desired Compile Time Architecture



API

SQL DML (*part of load*) (*independent system*) (*integration schema*)

SQL DML

Transaction Management

- Connect
- Begin transaction.
- Commit transaction.
- Abort transaction.
- Disconnect

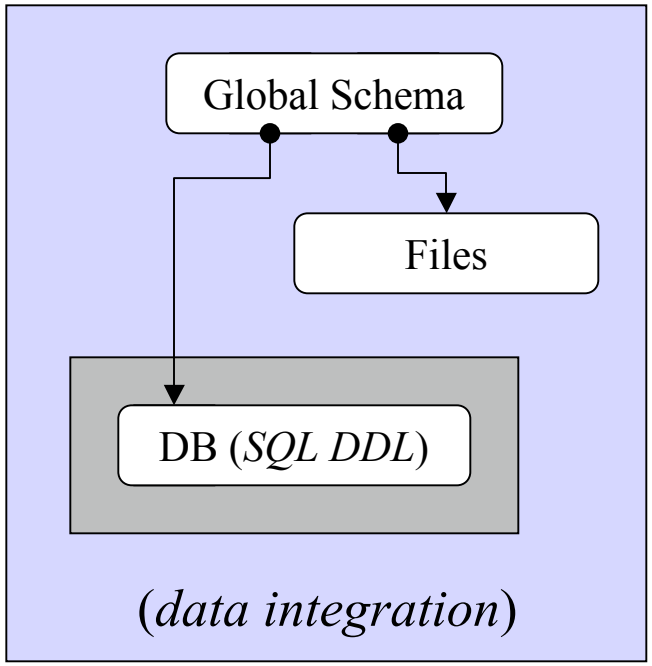
Data Access

- Access a tuple attribute value.
- Open an iterator (*defined by a static SQL query*).
- Succeed in accessing an iterator.
- Fail in accessing an iterator.
- Increment an iterator.
- Close an iterator.

Data Revision

- Update a tuple attribute value.
- Create a new tuple.
- Delete an existing tuple.

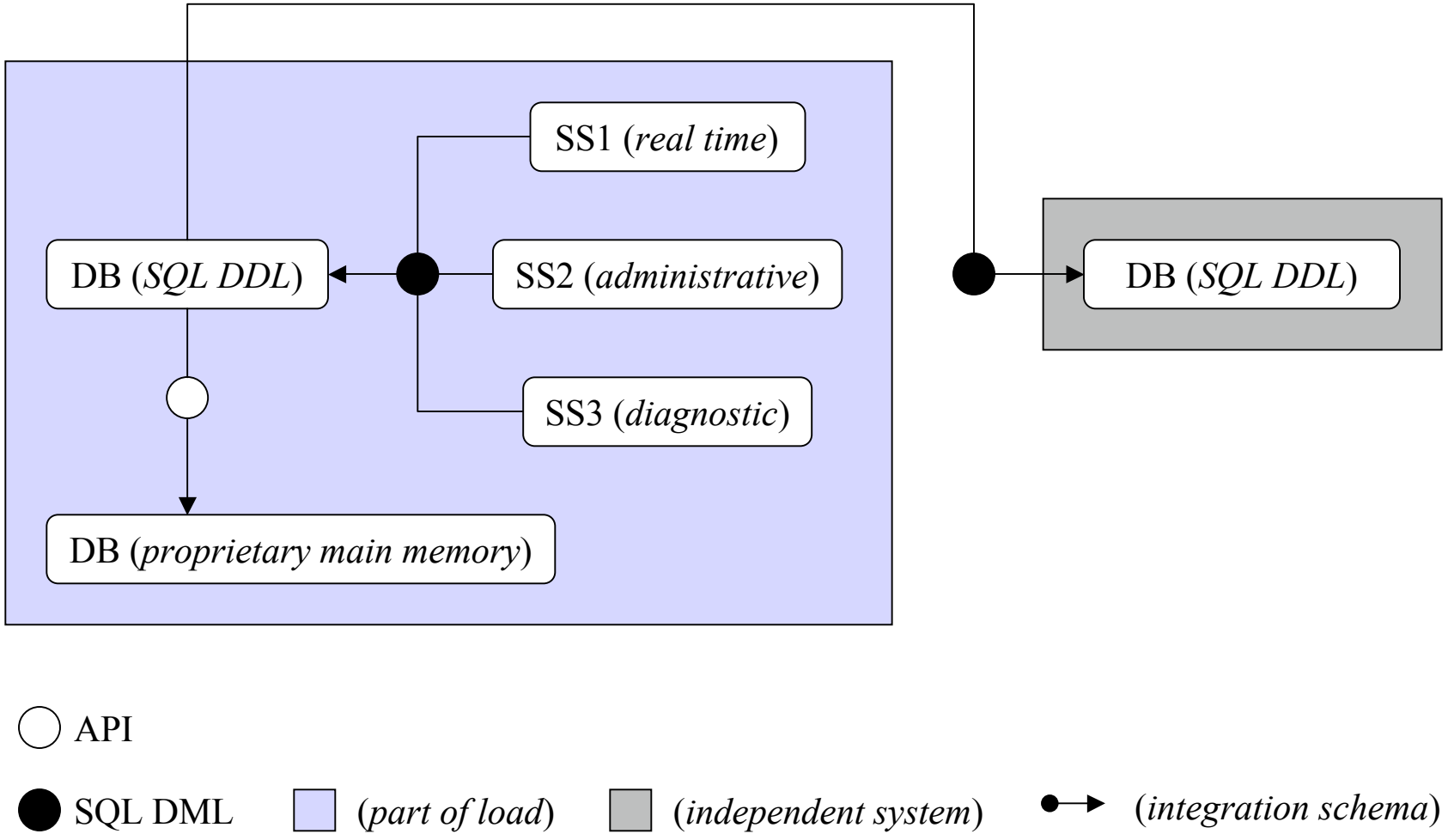
Integration in Heavyweight DB Engines



API

SQL DML (*part of load*) (*independent system*) (*integration schema*)

Lightweight Engines: Sybase Ultralight



SQL DML

Transaction Management

- Connect
- Begin transaction.
- Commit transaction.
- Abort transaction.
- Disconnect

Special Data Revision

- Synchronize

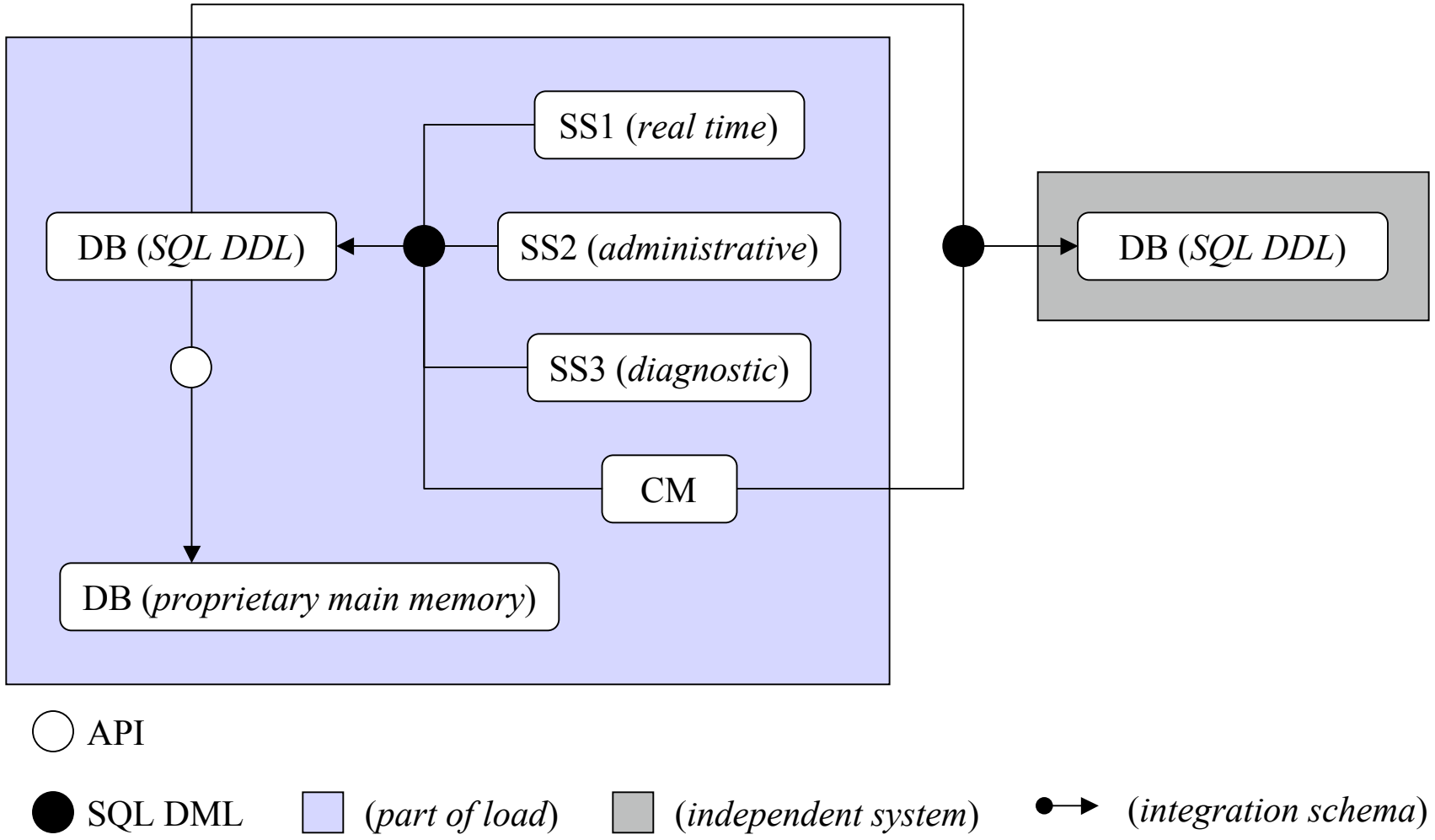
Data Access

- Access a tuple attribute value.
- Open an iterator
  (*defined by a static SQL query*).
- Succeed in accessing an iterator.
- Fail in accessing an iterator.
- Increment an iterator.
- Close an iterator.

Data Revision

- Update a tuple attribute value.
- Create a new tuple.
- Delete an existing tuple.

Main Memory Databases: TimesTen



Additional Operations Supported

Conceptual Data Revision

- Create table.
- Delete table.
- Create view.
- Delete view.

Physical Design

- Create index.
- Delete index.

Data Access

- Open an iterator
  (*defined by a SQL query string*).

A Performance Benchmark

The LINUX kernel.

Real time subsystem: `fork`, `malloc`, `open`, `connect`, ...

Administrative subsystem: `ps`, `ls`, ...

Competition: What expert C programmers can do in coding to the generic runtime architecture.

Issues in Query Optimization

- Pointers and arrays.
- Pipelined query plans.
- Code inlining.
- Semantic query optimization.
- Timing.
- Safety.

Issues in Concurrency Control

- Deadlock free protocols.
- Concurrency requirements.

Issues in Backup and Recovery

- Reliable main memory.
- Backup and recovery requirements.
- Compensating transactions.
- User specified recovery.

Issues in Physical Database Design

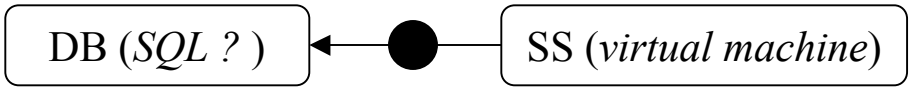
- Tuple identification.
- Field layout.
- Indexing: arrays, stacks, heaps, …†

†See, e.g., Knuth, volume 3.

Final Reflections

What is the main memory data?

- Anything in heap memory.
- Anything on execution stacks?
- The application code?



SQL DML