

Distributed Database

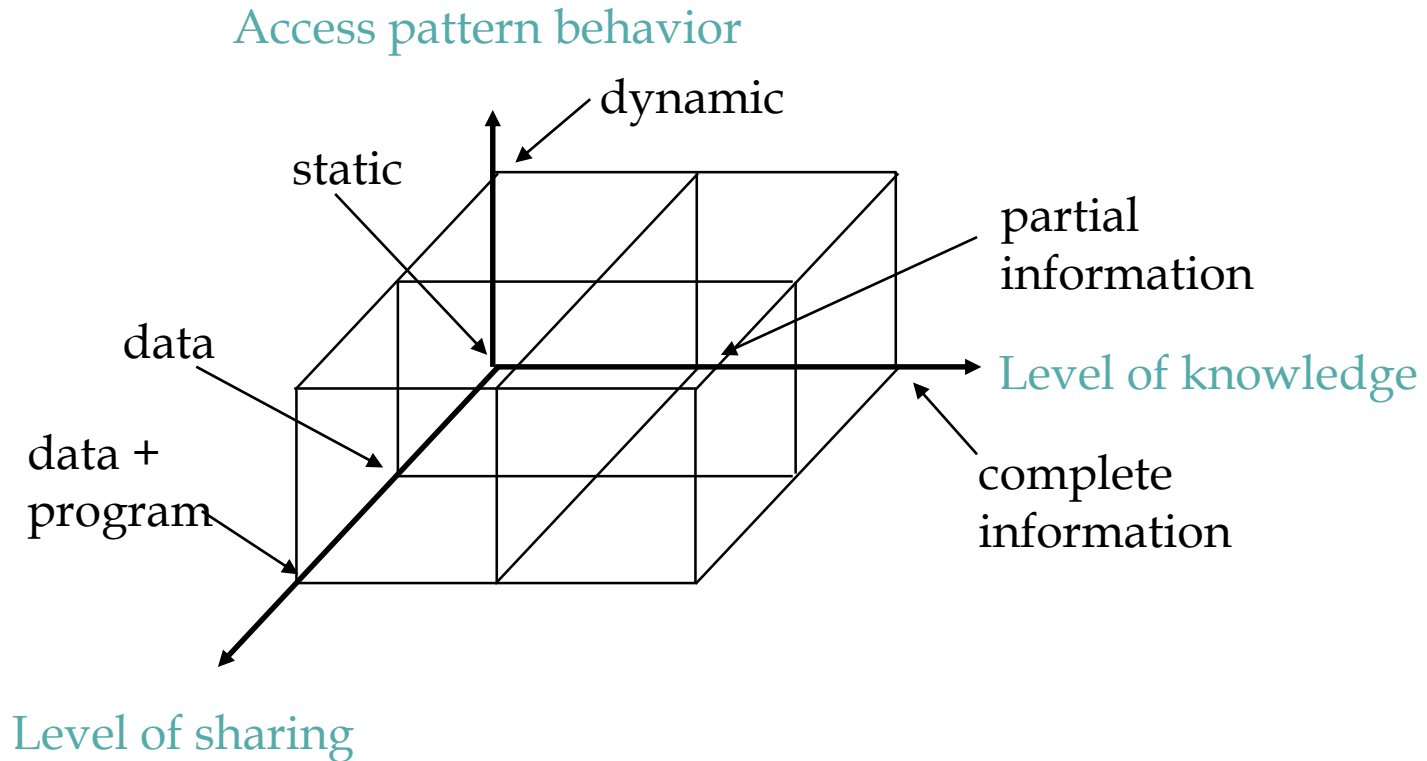
Munawar, PhD



Bottom Up Design



Dimensions of the Problem





Distributed Database Design

Level of sharing

- ❖ no sharing - each application and its data execute at one site,
- ❖ data sharing - all the programs are replicated at all the sites, but data files are not,
- ❖ data plus program sharing - both data and programs may be shared.

Behavior of access patterns

- ❖ static - access patterns of user requests do not change over time,
- ❖ dynamic - access patterns of user requests change over time.

Level of knowledge on access pattern behavior

- ❖ complete information - the access patterns can reasonably be predicted and do not deviate significantly from the predictions,
- ❖ partial information - there are deviation from the predictions.



Top-Down vs. Bottom-Up Design

- ❖ Database applications, like other complex systems, can be designed by either of two strategies, or approaches, called the "top-down" strategy and the "bottom-up" strategy.
- ❖ In practice, a designer or design team rarely uses solely one or the other of these strategies. Nevertheless, their differences are worth noting.
- ❖ **Top-Down Design**
 - ❑ This approach starts by identifying the entities in the problem and their relationships, and then working down to the attributes and their details.
- ❖ **Bottom-Up Design**
 - ❑ This approach starts with the attributes in the problem, and then works up by identifying the entities to which the attributes need to be linked and, in turn, the relationships among the entities



Top-Down vs. Bottom-Up Design (continued)

- ❖ Rob and Coronel comment quite carefully about these two somewhat different strategies, saying:
 - ❑ "Although these two methodologies are complementary rather than mutually exclusive, a primary emphasis on a bottom-up approach *may be* [emphasis added] more productive for small databases with few entities, attributes, relations, and transactions. For situations in which the number, variety, and complexity of entities, relations, and transactions is overwhelming, a primarily top-down approach *may be* [emphasis added] more easily managed."



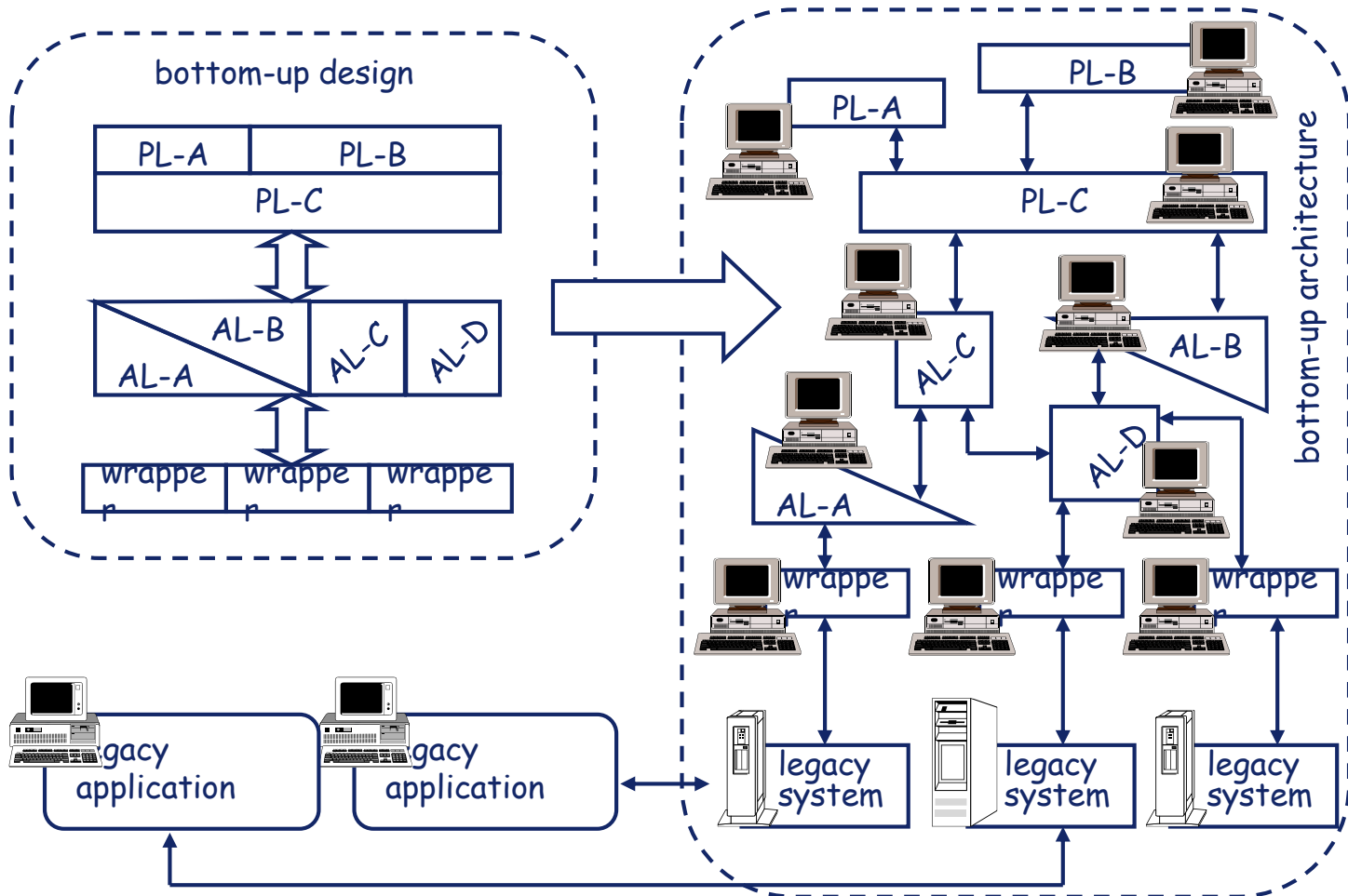
Date's Twelve Rules for Distributed Database Systems



Sumber informasi <http://www.exploredatabase.com/2015/03/dates-twelve-rules-for-distributed-database.html>



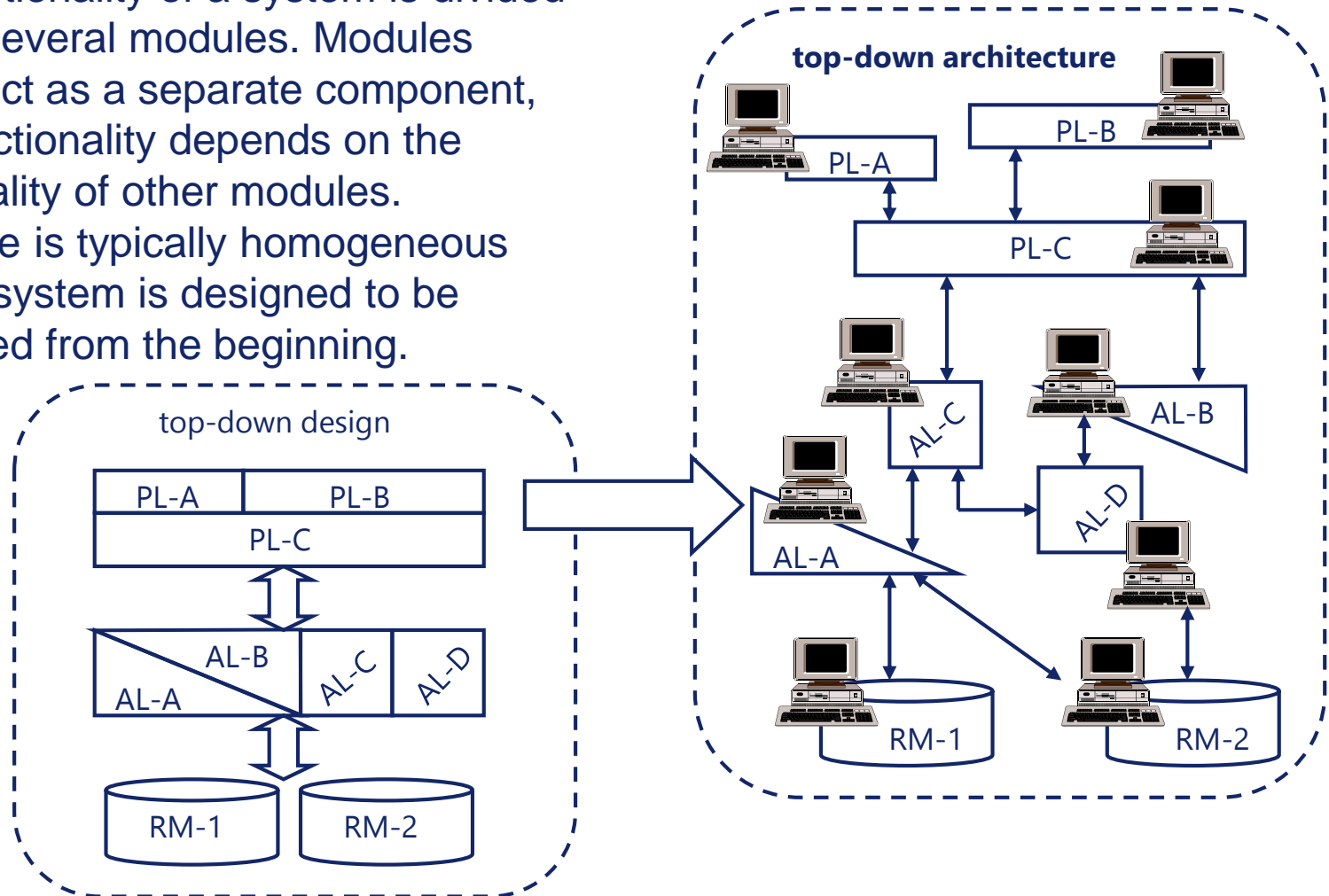
Bottom Up design





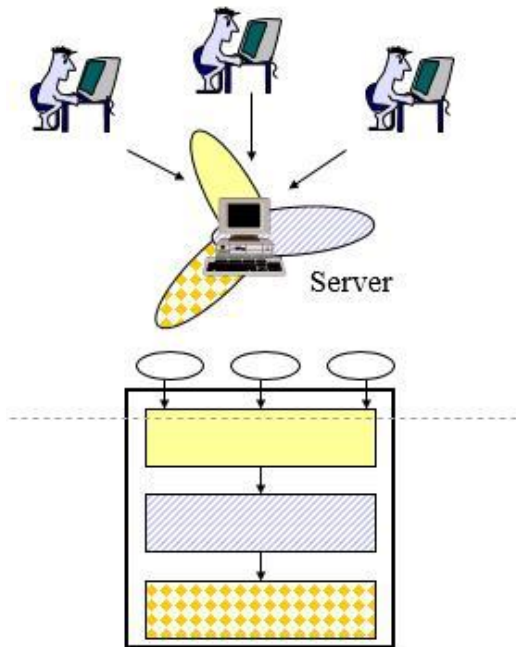
Top down design (review)

- ❖ The functionality of a system is divided among several modules. Modules cannot act as a separate component, their functionality depends on the functionality of other modules.
- ❖ Hardware is typically homogeneous and the system is designed to be distributed from the beginning.

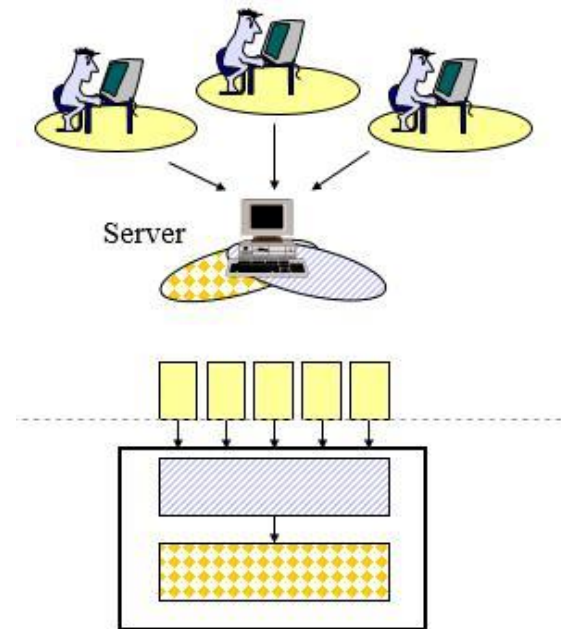




One tier: fully centralized

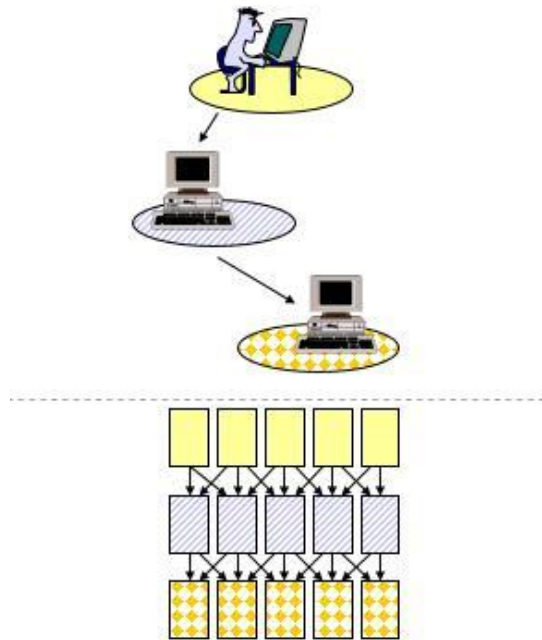


Two tier: client/server

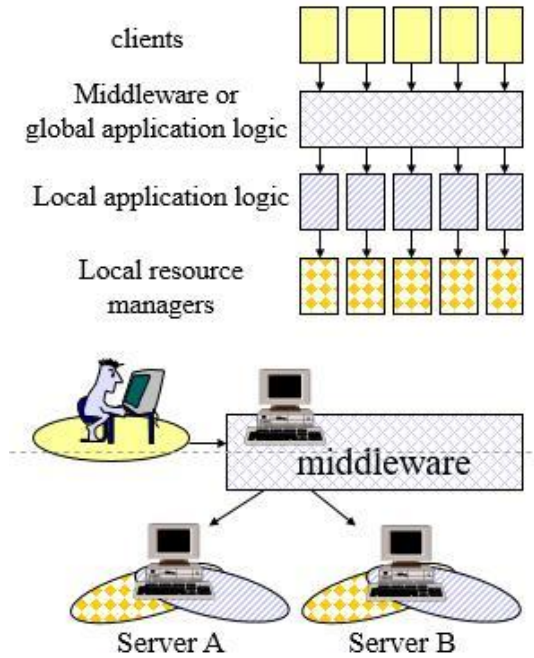




Three tier: middleware



Middleware





Analysis of a Top-Down Bottom-Up Data Analysis Framework and Software Architecture Design

Paper (bahan diskusi) tittle *Distributed Database Design Bottom Up ...* link
url <http://web.mit.edu/smadnick/www/wp/2014-08.pdf>

Thank You !

Munawar, PhD

