合肥學院
HEFEI UNIVERSITY


# Distributed Computing
## Lesson 15: 3-Tier Application Structure

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://www.it-weise.de

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号

# Outline

1 3-Tier Application Structure

# Overview

- What are 3-tier architectures?
- How can they be constructed with the technologies we already learned?

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5]

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier:
     - business logic and business objects

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier:
     - business logic and business objects
     - maybe accessible via RPC or RMI or Java Servlets

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier:
     - business logic and business objects
     - maybe accessible via RPC or RMI or Java Servlets
     - business logic $\longrightarrow$ does *not* belong into a JSP...

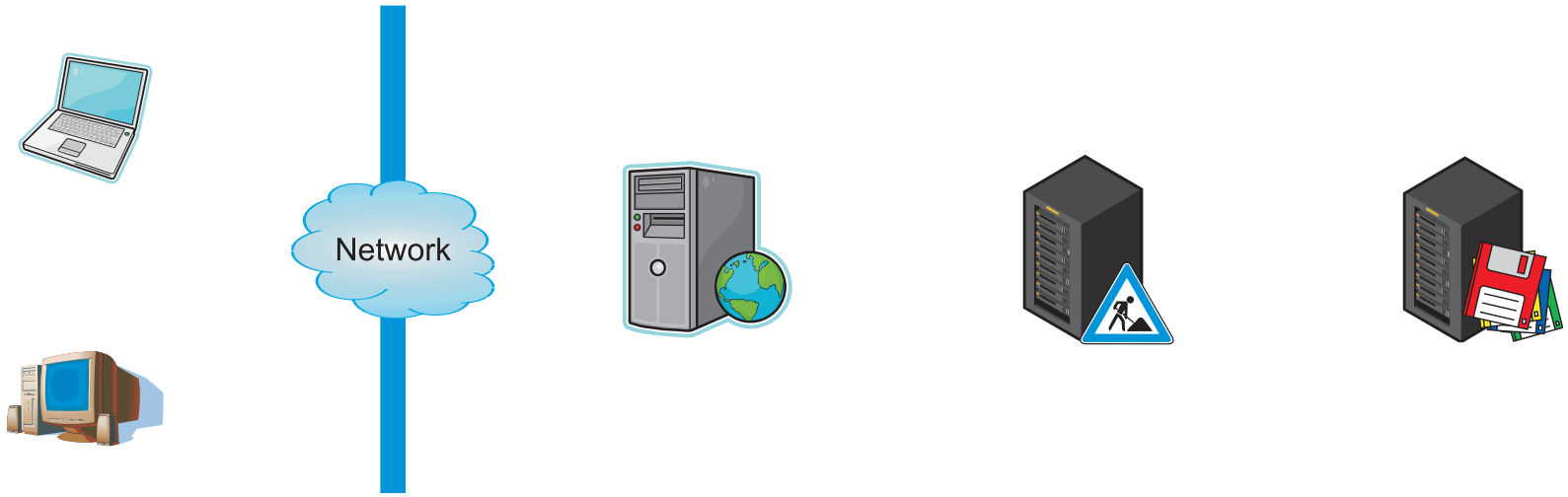
# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier:
     - business logic and business objects
     - maybe accessible via RPC or RMI or Java Servlets
     - business logic $\longrightarrow$ does *not* belong into a JSP...
  3. Data/Persistency Tier

# Tiered Architectures

- Goal: Separation of concerns (SoC) [1–3]
- Application divided into several layers [4, 5], e.g.,
  1. Presentation Tier:
     - user interface / front end, e.g., web pages, JSP
     - representation of data
     - input from the user
  2. Application Tier:
     - business logic and business objects
     - maybe accessible via RPC or RMI or Java Servlets
     - business logic $\longrightarrow$ does *not* belong into a JSP...
  3. Data/Persistency Tier:
     - database server

# 3-Tier Architectures



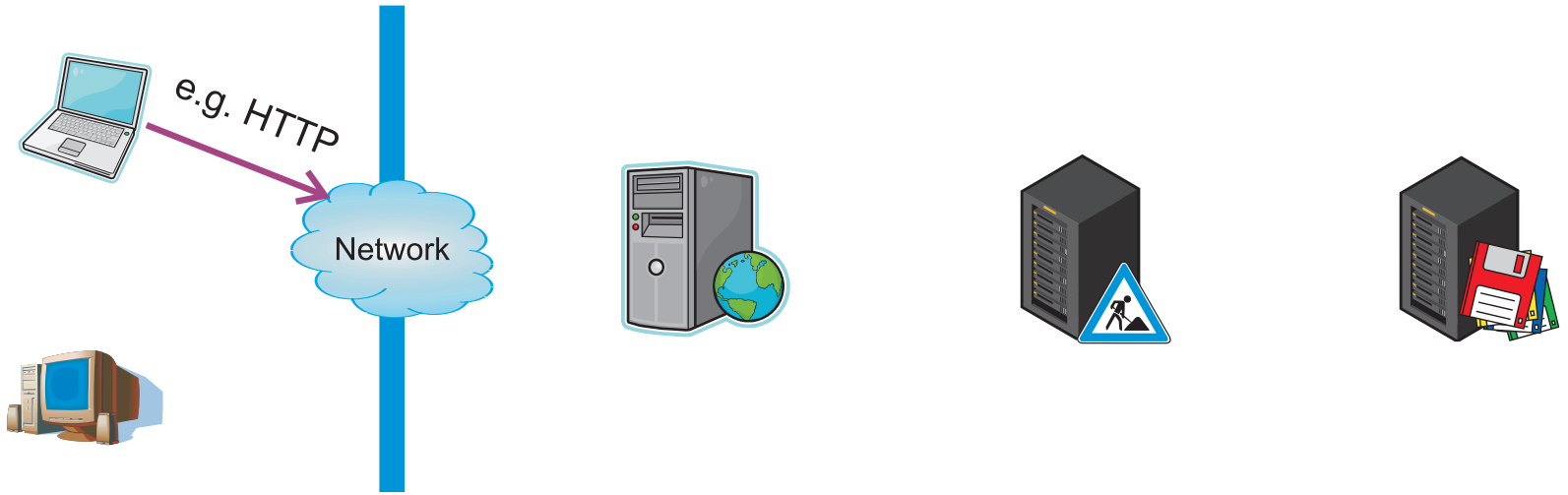

Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

# 3-Tier Architectures



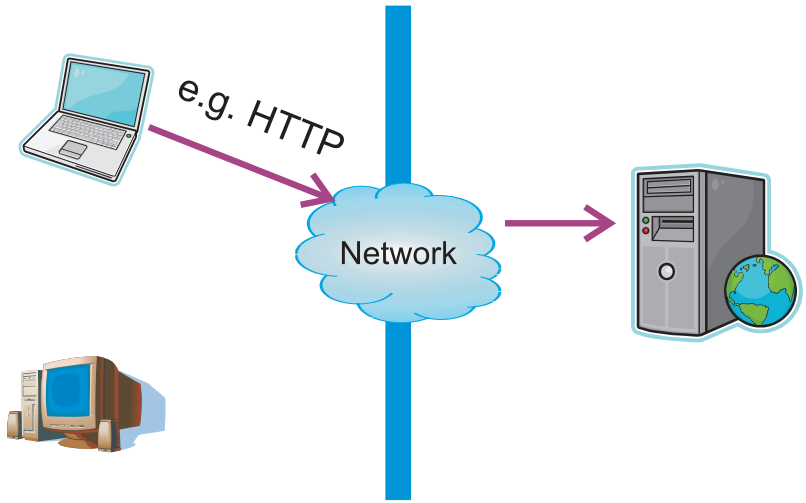

Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

# 3-Tier Architectures



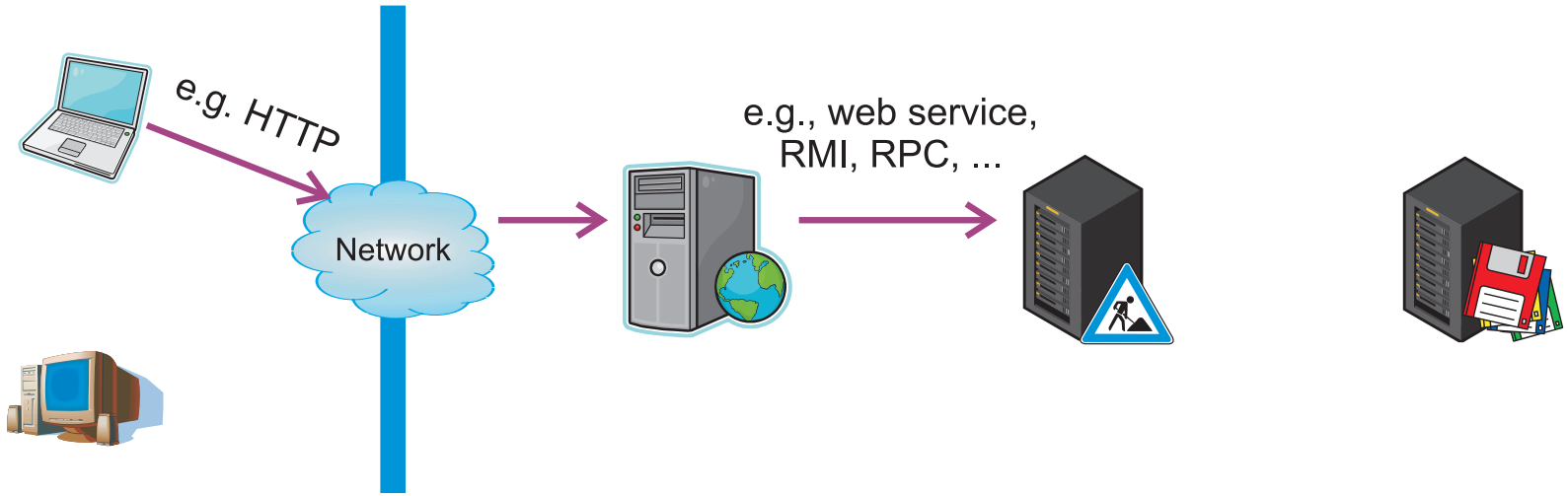

Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

# 3-Tier Architectures



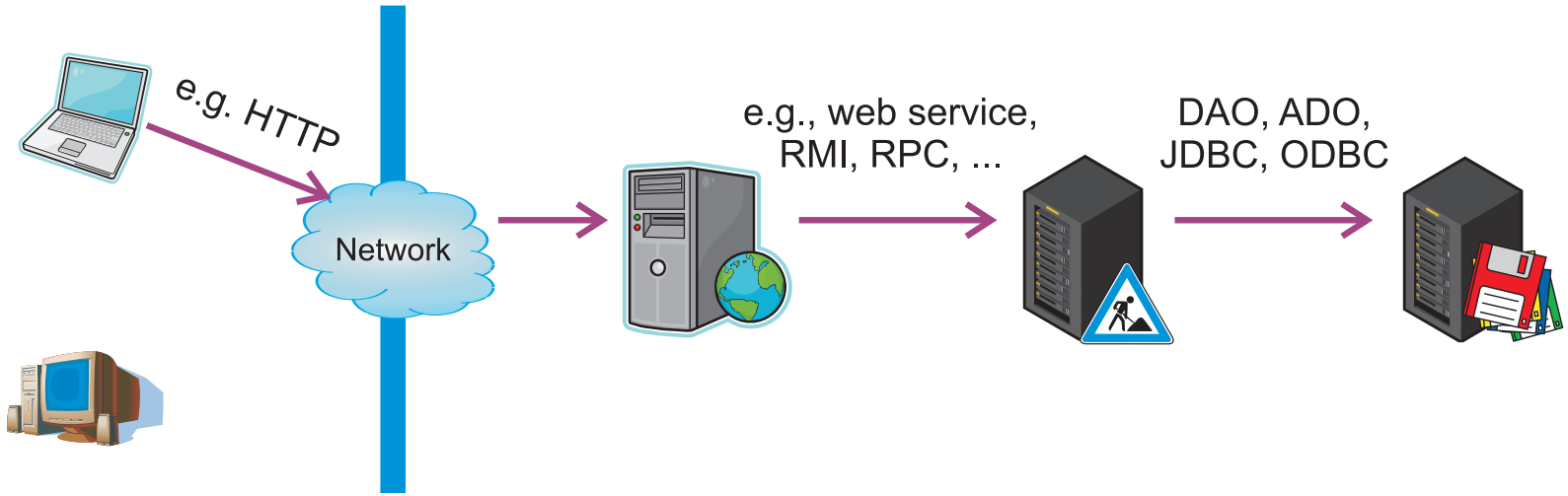

Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

# 3-Tier Architectures



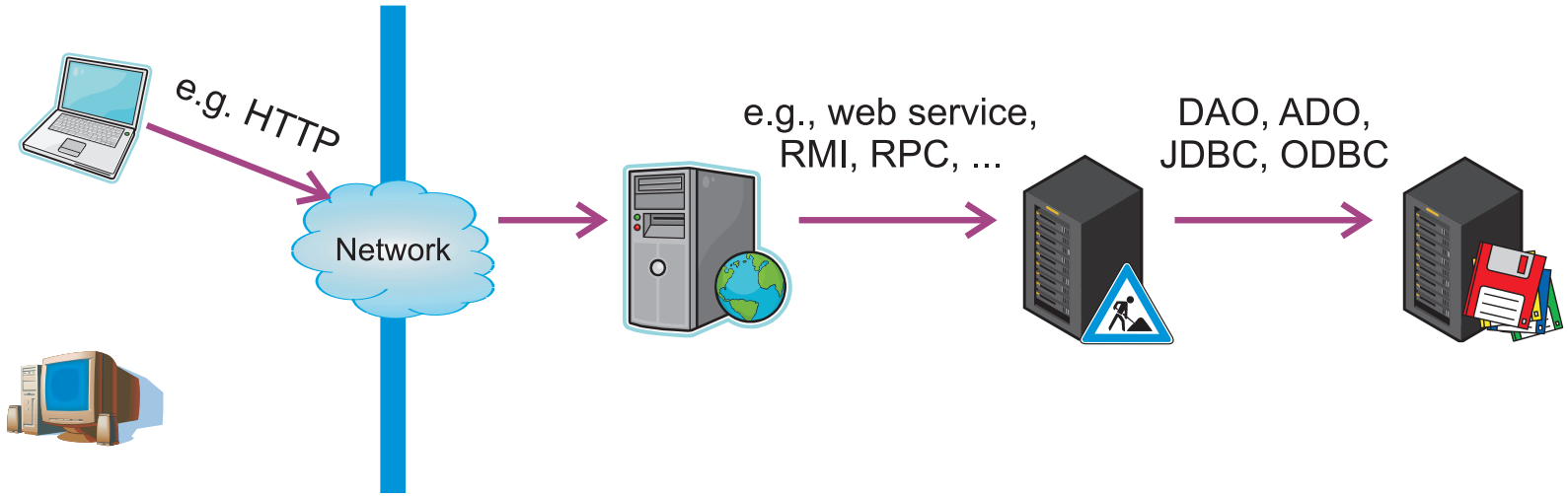

Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

# 3-Tier Architectures




Clients
(e.g., web browsers)

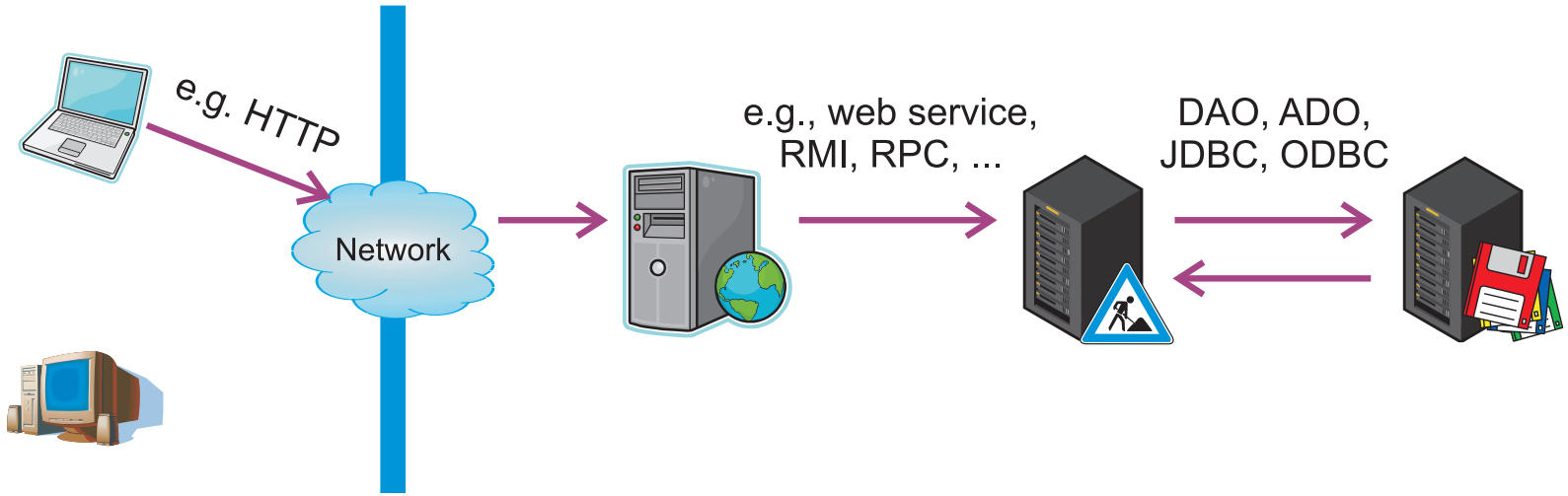
Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

Database Manager
knows about performance, savety, redundancy, backup of data storage

# 3-Tier Architectures




Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

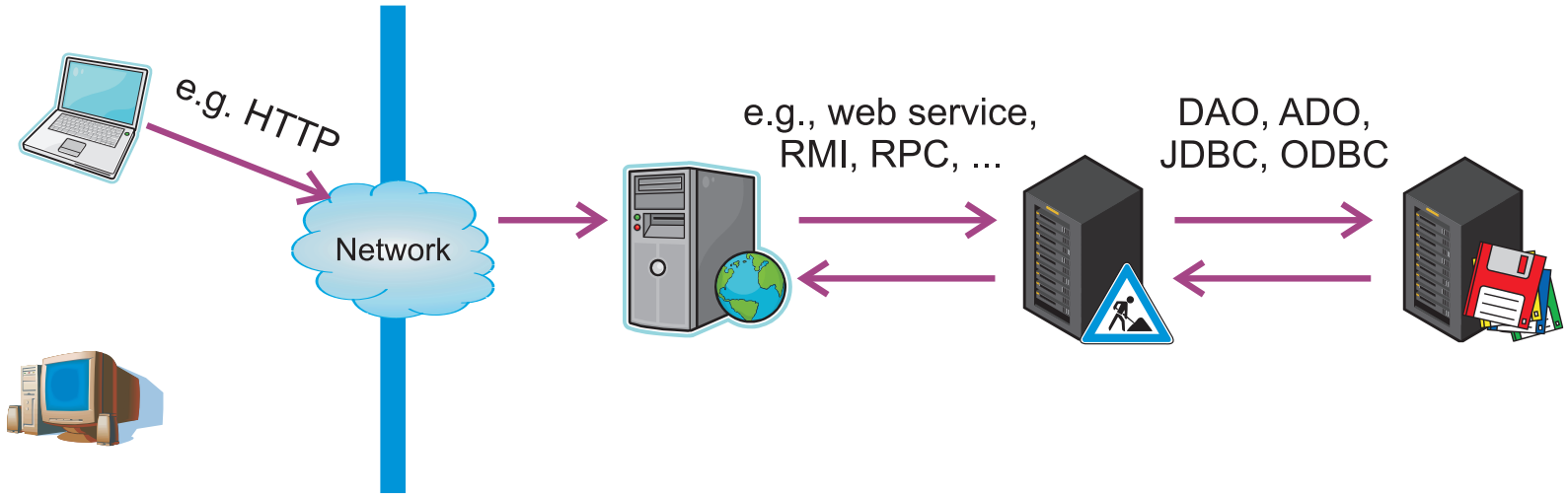
Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

**Application Designer:**
knows business logic and rules; cooperates with business departments

**Database Manager**
knows about performance, savety, redundancy, backup of data storage

# 3-Tier Architectures




Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

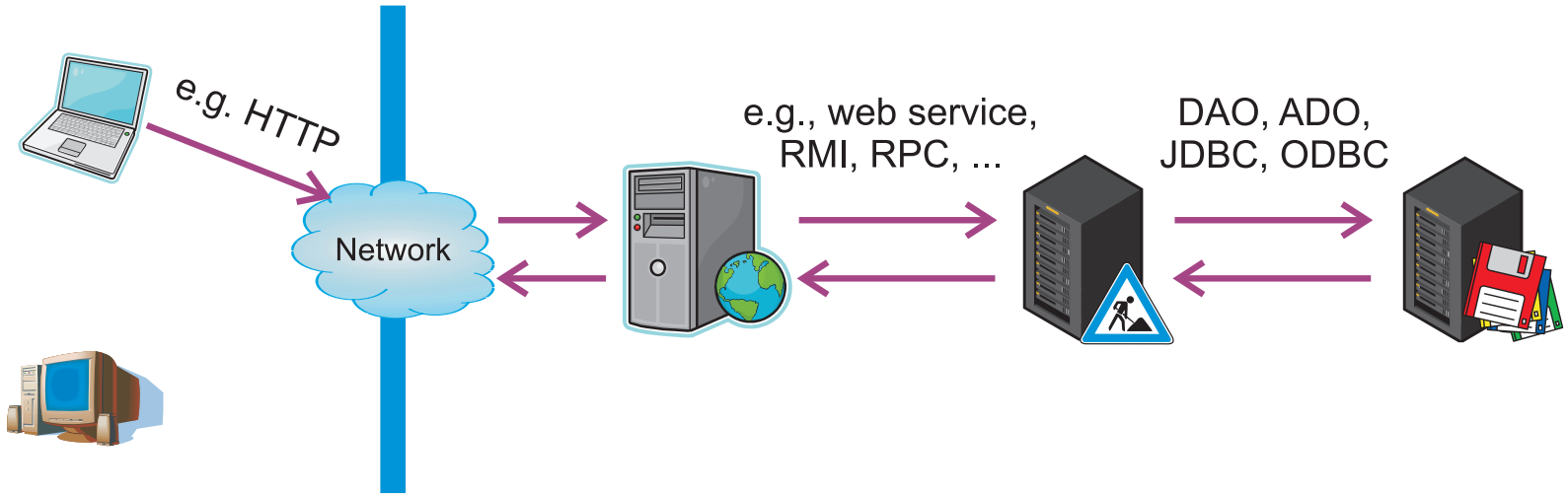
Persistency Layer
(e.g., database server)

**Web Designer:**
knows about ergonomics and design; cooperates with users

**Application Designer:**
knows business logic and rules; cooperates with business departments

**Database Manager**
knows about performance, savety, redundancy, backup of data storage

# 3-Tier Architectures




Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

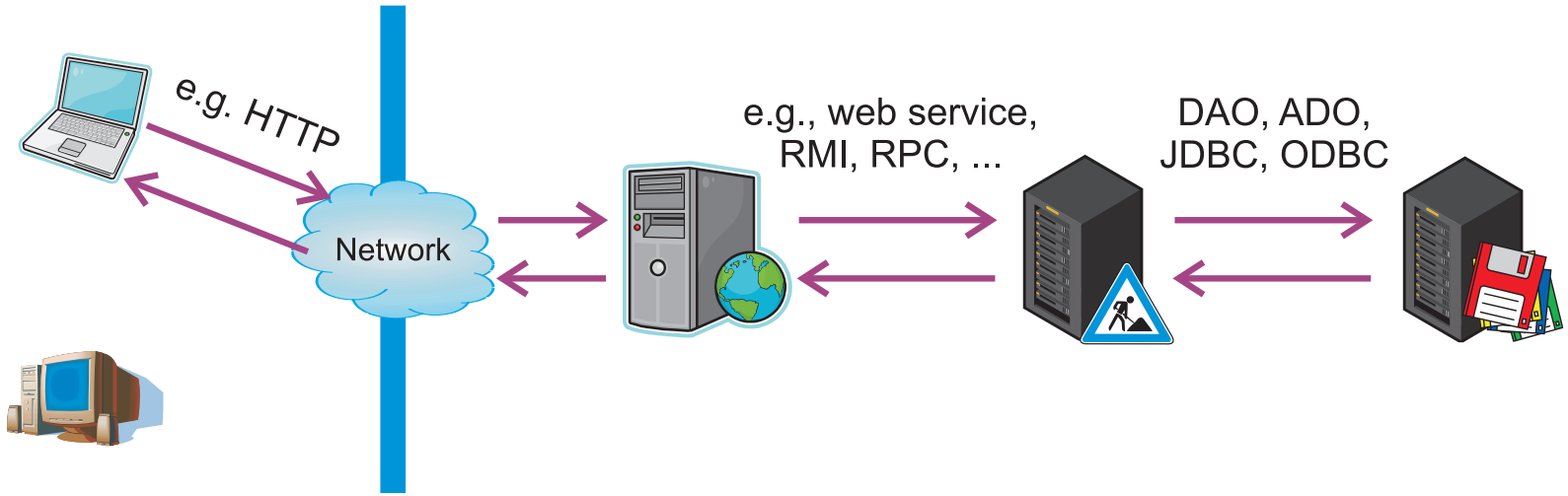
Persistency Layer
(e.g., database server)

**Web Designer:**
knows about ergonomics and design; cooperates with users

**Application Designer:**
knows business logic and rules; cooperates with business departments

**Database Manager**
knows about performance, savety, redundancy, backup of data storage

# 3-Tier Architectures




Clients
(e.g., web browsers)

Front End/Presentation
(e.g., web server)

Business/Appl. Logic
(Application Server)

Persistency Layer
(e.g., database server)

**Web Designer:**
knows about ergonomics and design; cooperates with users

**Application Designer:**
knows business logic and rules; cooperates with business departments

**Database Manager**
knows about performance, savety, redundancy, backup of data storage

# Why?

## Colored Logic 2005-08-31 http://thedailywtf.com/Articles/Colored_Logic.aspx

*by Alex Papadimoulis in Feature Articles (136 Comments)*

When developing software that requires a user interface, some of us strive for a Model-View-Controller architecture to ensure ease of change to the UI and data components. Others will couple their business and display logic a little more tightly, using variables and classes to go between the two. And then there are the few (**Adam Courtney's** colleague included) who use the color of a UI component to determine business logic and program flow ...

```
If lblClientName.BackColor = RGB(255, 0, 0) Then 'new client

    'Setup the client account
    Call SaveNewClientData

    'notify use of success
    If lblClientNum.ForeColor = RGB(0, 0, 153) Then
        MsgBox "Client was created but is pending activation."
    ElseIf lblClientNum.ForeColor = RGB(0, 128, 0) Then
        MsgBox "Client was created and was successfully activated."
    ElseIf lblClientNum.ForeColor = RGB(255, 0, 0) _
            And lblClientNum.Caption = "999999" Then
        MsgBox "There was an error creating the client. " & _
               "Please ensure all required fields were entered."
    End If

    'ED: Snip ...
End If
```

source: [6]

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures.

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications
- Disadvantages

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications
- Disadvantages:
  - Structure a bit more complex compared to single/monolithic systems

# Pros and Cons

- Name some advantages/disadvantages of 3-tier application structures [3].
- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications
- Disadvantages:
  - Structure a bit more complex compared to single/monolithic systems
  - Physical separation of business logic and database decreases performance compared to single/monolithic systems

# Pros and Cons

- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications
- Disadvantages:
  - Structure a bit more complex compared to single/monolithic systems
  - Physical separation of business logic and database decreases performance compared to single/monolithic systems
- 3-tier architecture used in many small- and mid-sized business infrastructures

# Pros and Cons

- Advantages:
  - Business logic removed from: database server, presentation layer, and client $\Longrightarrow$ better performance, maintainence
  - Complex application rules easy to implement in application server
  - Changes to business logic automatically enforced by server – changes require only new application server software to be installed
  - Application server logic is portable to other database server platforms
  - Presentation layer holds no business logic
  - if application tier is presented via RPC, RMI, or as Servlets, it becomes easier to later integrate functionality with other application tiers of other, related applications
- Disadvantages:
  - Structure a bit more complex compared to single/monolithic systems
  - Physical separation of business logic and database decreases performance compared to single/monolithic systems
- 3-tier architecture used in many small- and mid-sized business infrastructures
- RMI can be used to build such an infrastructure

# Summary

- 3-tier architectes allow clear separation of concerns
- they allow us to build extensible web applications
- RMI is one way to realize the communication between the front end and the middle tier in a 3-tier architecture

# 谢谢

# Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://www.it-weise.de

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog

# Bibliography

# Bibliography I


1. Edsger Wybe Dijkstra. On the role of scientific thought. Technical report, The Netherlands, Nuenen: Burroughs Corporation, August 30, 1974. URL http://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html.
2. Edsger Wybe Dijkstra. On the role of scientific thought. In *Selected Writings on Computing: A Personal Perspective*, Texts and Monographs in Computer Science, pages 60–65. Berlin, Germany: Springer-Verlag GmbH, 1982.
3. Channu Kambalyal. *3-Tier Architecture*. Round Lake, IL, USA: Sushil Consultants Inc. – An Information Technology & Management Services company, May 26, 2005. URL http://channukambalyal.tripod.com/NTierArchitecture.pdf.
4. Jeri Edwards. *3-tier Client/Server at Work*. New York, NY, USA: John Wiley & Sons Ltd., 1999. ISBN 0471315028 and 9780471315025. URL http://books.google.de/books?id=N1MAQAAIAAJ.
5. Paul D. Sheriff. *Fundamentals of N-Tier Architecture*. Scottsdale, AZ, USA: PDSA – Solutions for the Real World, 2006. ISBN 0979374812 and 9780979374814. URL http://books.google.de/books?id=0IHZPQAACAAJ.
6. Alex Papadimoulis. Colored logic. *The Daily WTF – Curious Perversions in Information Technology*, August 31, 2005. URL http://thedailywtf.com/Articles/Colored_Logic.aspx.