Introduction to software architecture

Bo Simonsen

Department of Computer Science

University of Copenhagen

Recommended reading

- Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy, *Software Architecture: Foundations, Theory, and Practice*, Wiley (2009), §1,2,**3-6**.

- Jyrki Katajainen and Bo Simonsen, Applying design patterns to specify the architecture of a generic program library, *Foundations of an Adaptable Container Library*, M.Sc. Thesis by Bo Simonsen, Department of Computer Science, University of Copenhagen (2009).

What is software architecture

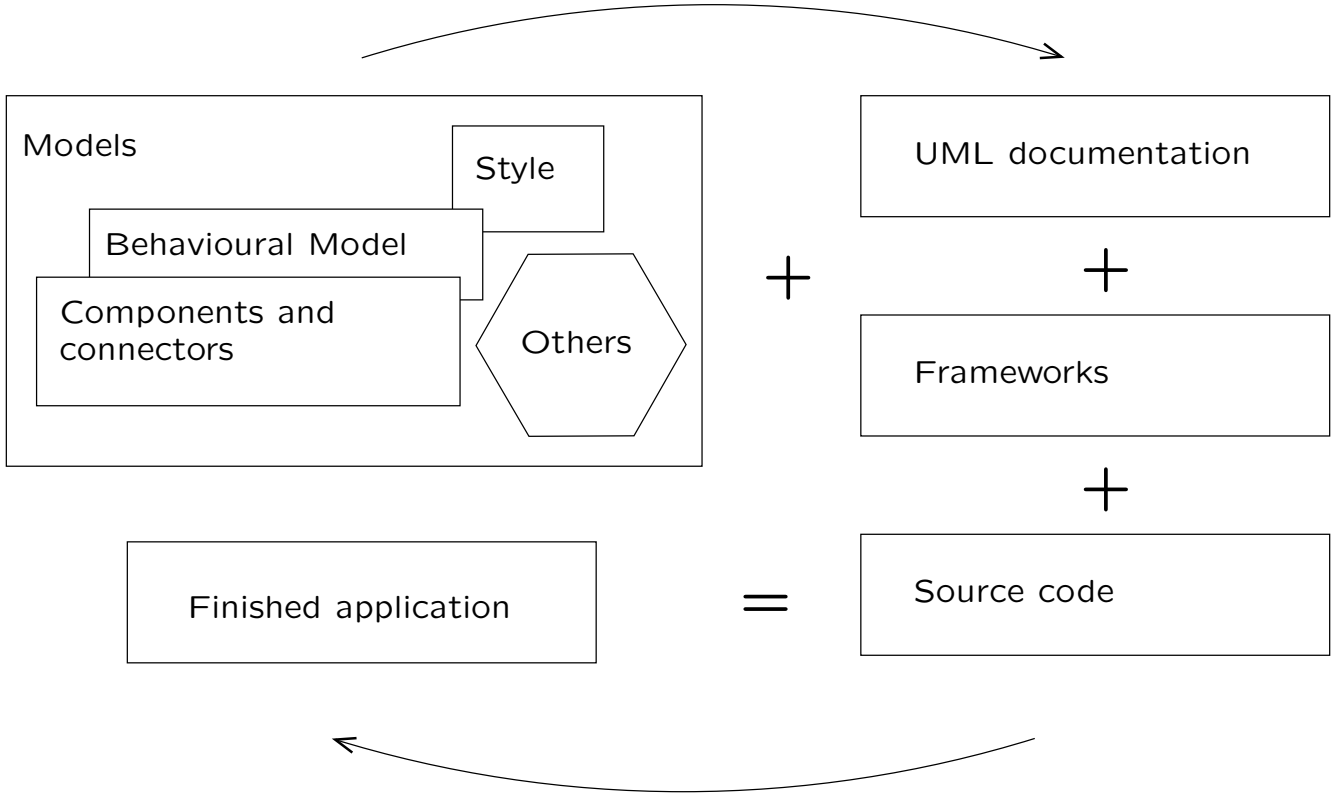
Software architecture means to use all your knowledge about programming (concurrent, OO, generic), libraries, operating systems, and networks to make the high-level design decisions about a system.

Fundamental facts:

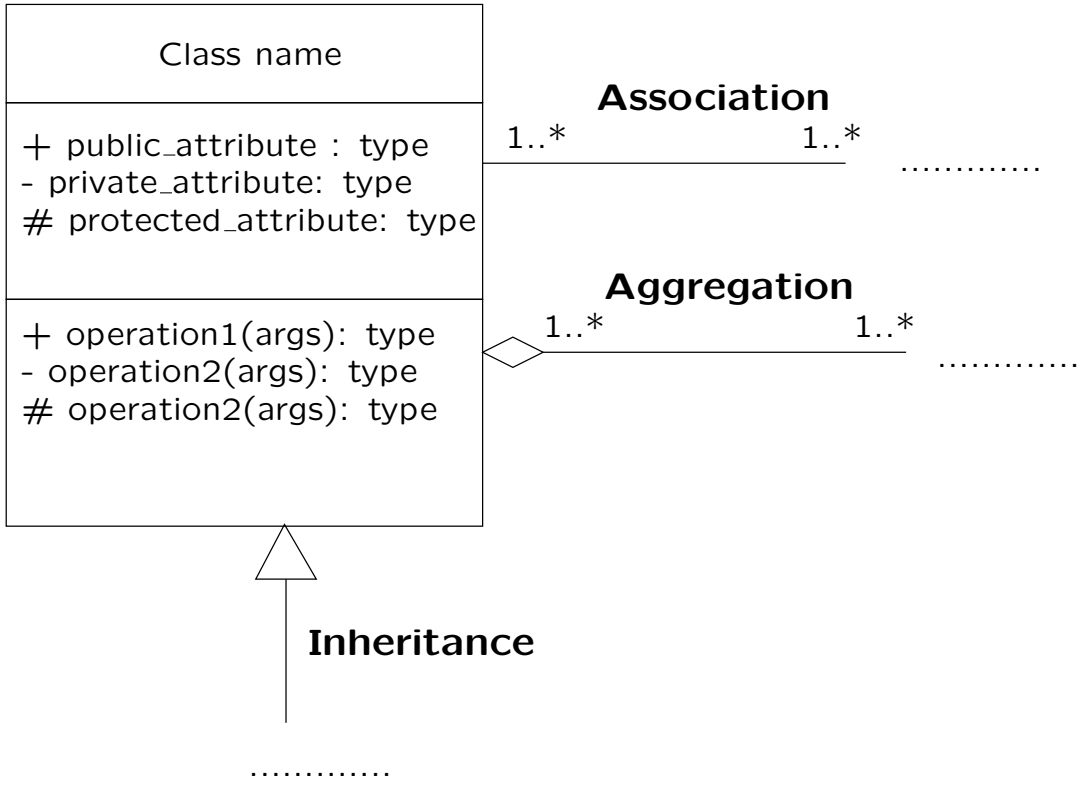
- Every application has an architecture.
- Every application has at least one architect.
- Architecture is **not** a phase of development.

Formal definition of software architecture

Definition: A software system's architecture is the set of principal design decisions made about the system.



UML Diagrams



Frameworks

Software frameworks: A software framework is a re-usable design for a software system (or subsystem). A software framework may include support programs, code libraries, a scripting language, or other software to help develop and glue together the different components of a software project. Various parts of the framework may be exposed through an API.

Source: *Wikipedia*.

Basic software architectural concepts

- Components,
- Connectors,
- Configuration,
- Styles,
- and Patterns.

Components

Definition: A software component is an architectural entity that

- encapsulates a subset of the system's functionality and/or data,
- restricts access to that subset via an explicitly defined interface, and
- has explicitly defined dependencies on its required execution context.

Connectors

Definition: A software connector is an architectural element tasked with effecting and regulation interactions among components.

Examples:

- Procedure call,
- Shared data access,
- Distribution (RPC),
- and Adaptor.

Case study I: UNIX Command interpreter

Consider the following expression given to a unix command interpreter (bash, tcsh, zsh, ..)

```
lspci|sed -n '/Network/p;/Ethernet/p;'|sed 's/.*: //'
```

What happens?

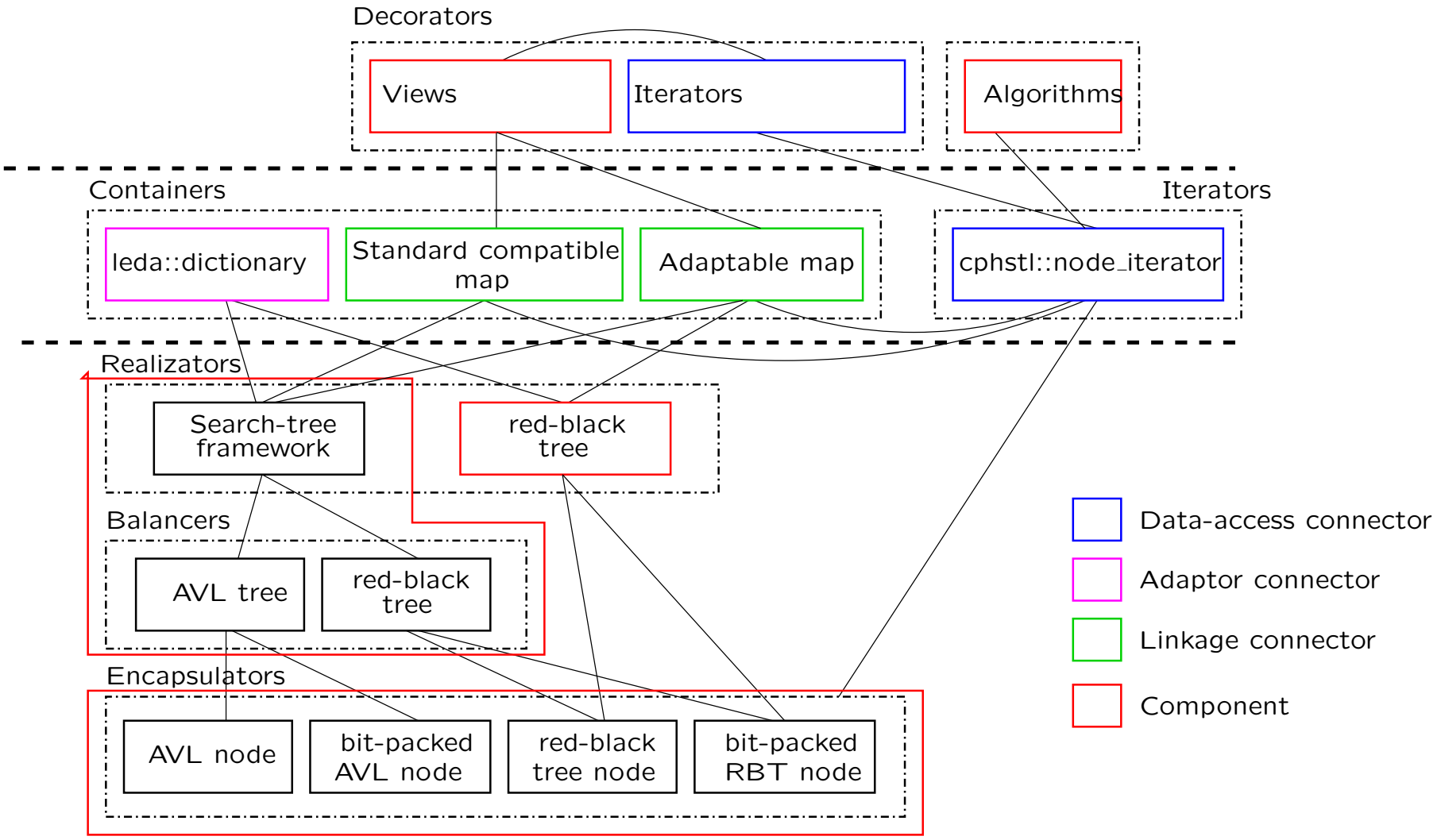
- The output of `lspci` is given to `sed`,
- `sed` selects only the lines containing Network or Ethernet.
- The output of the sed command is used in another sed command to remove every before `:`.

We can view the tools (lspci, sed) etc. as components. These components are connected using the | token.

Case study II: The CPH STL

- The Copenhagen STL is an enhanced version of the STL developed here at DIKU.
- The STL provides: Algorithms, Containers, and Iterators.
- Details of the architecture of the CPH STL is described in my Master's Thesis.

The architecture of the CPH STL



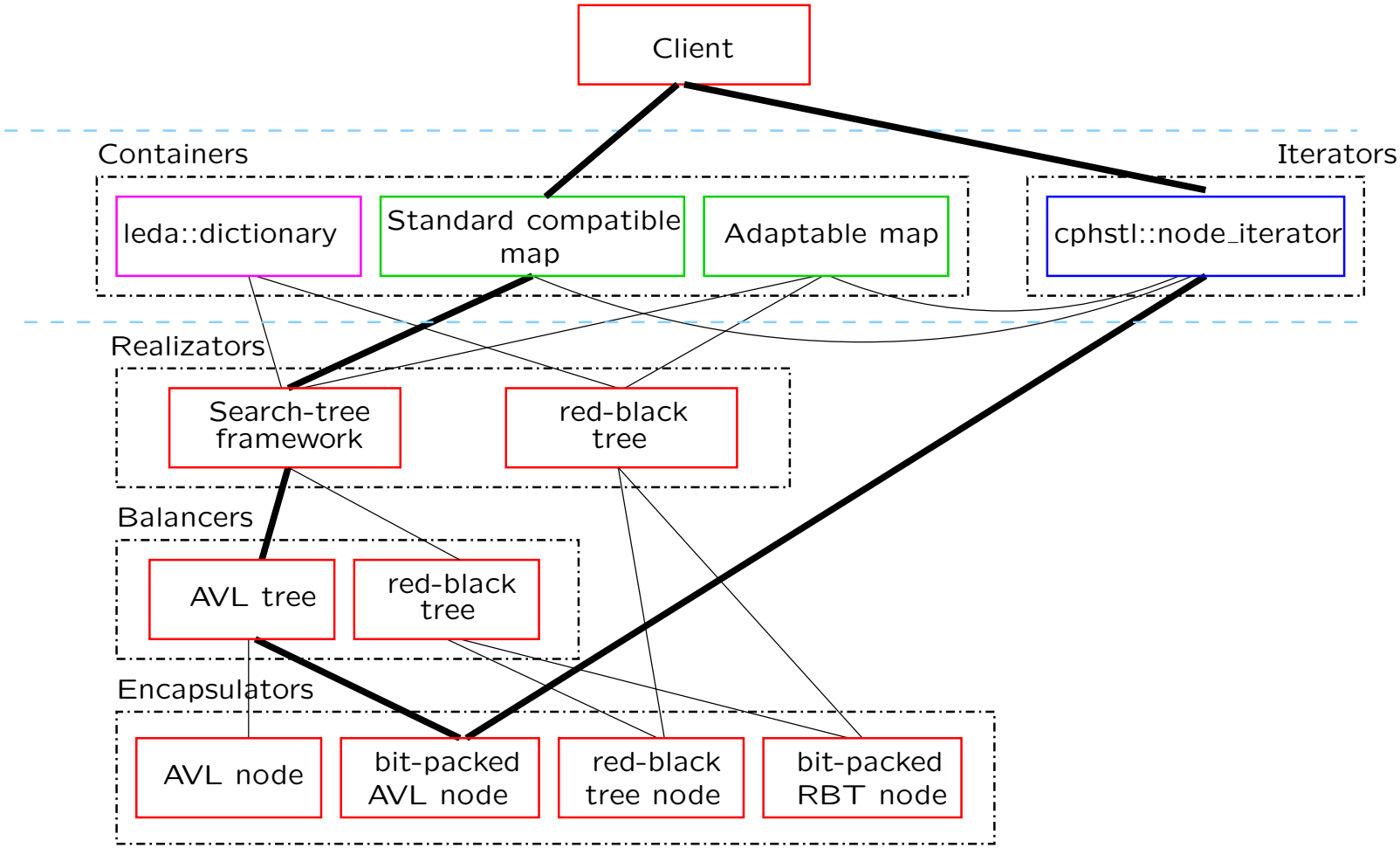
A configuration

Definition: An architectural configuration is a set of specific associations between the components and the connectors of a software system's architecture.

Example: The command given to the UNIX command interpreter as we saw before:

```
lspci|sed -n '/Network/p;/Ethernet/p;'|sed 's/.*: //'
```

A configuration of the CPH STL

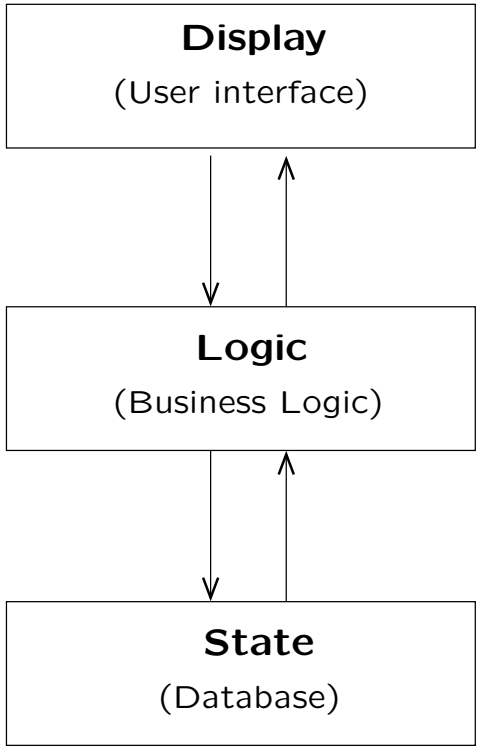


Patterns

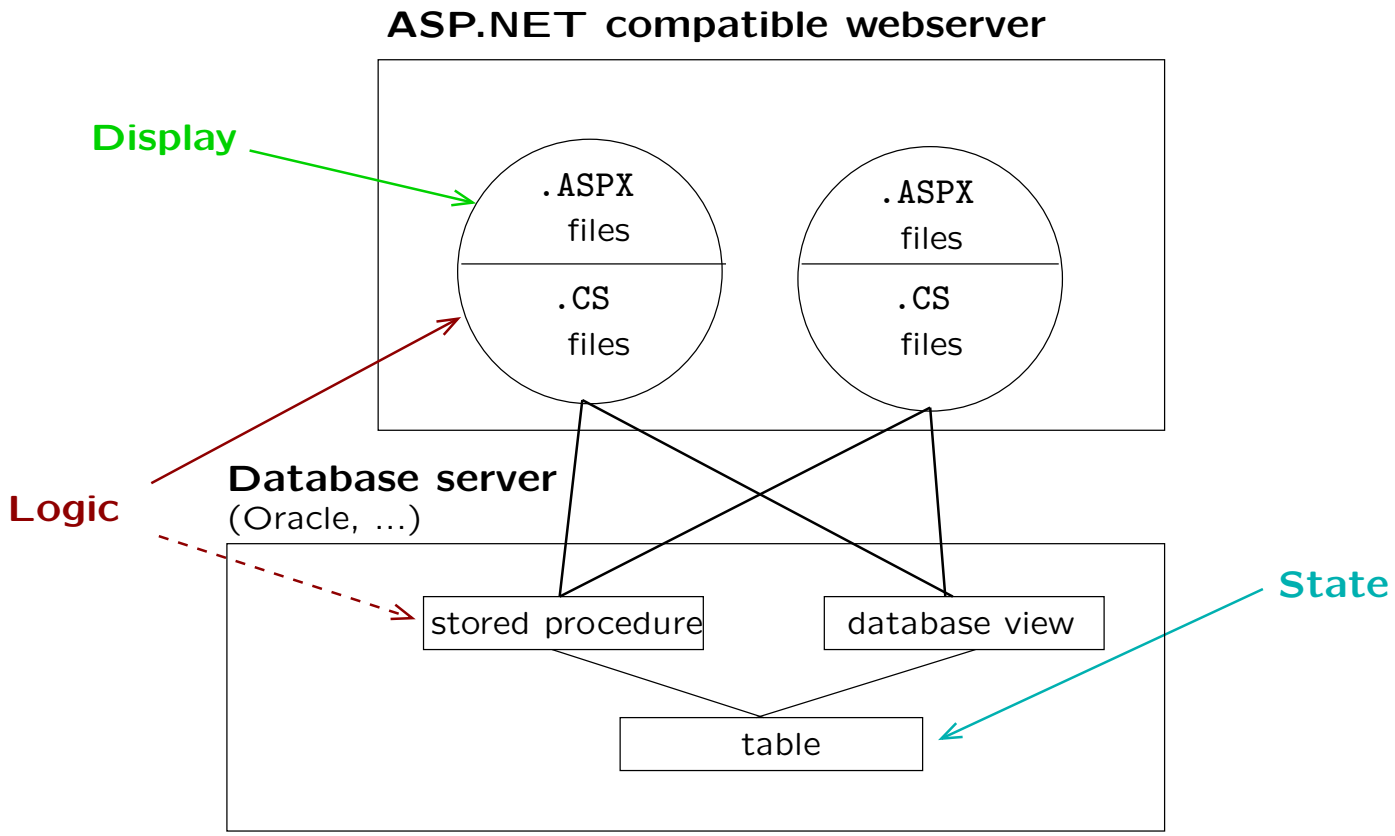
Definition: An architectural pattern is a named collection of architectural design decisions that are applicable to a recurring design problem, parameterized to account for different software development contexts in which that problem appears.

The definition for design patterns is similar, architectural patterns are just high-level patterns.

State-Logic-Display



Example: Web-server applications



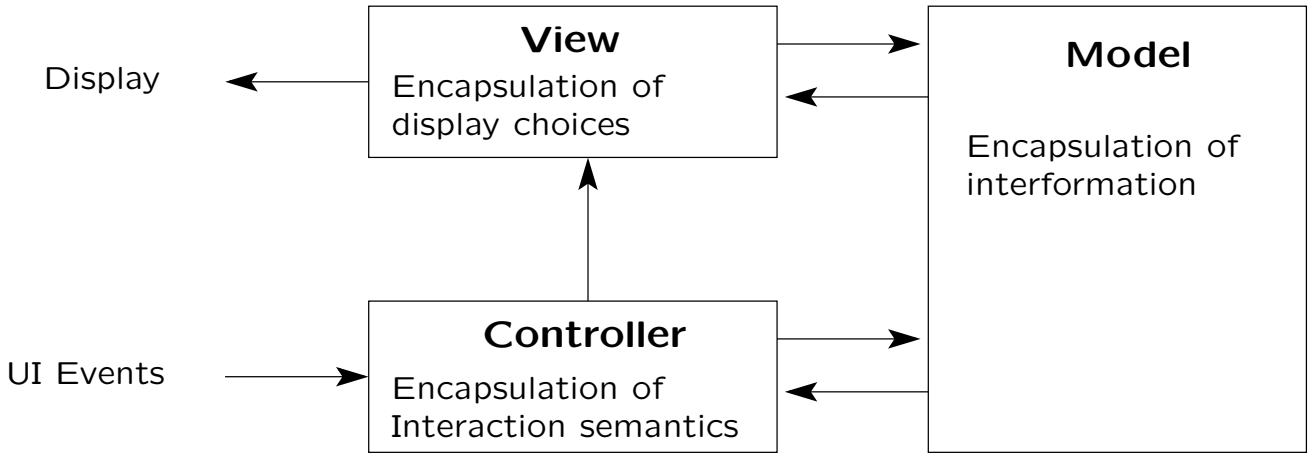
Sample ASPX code

```
<%@ Page Language="C#" Inherits="TestWeb.Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org
<html>
<head>
        <title>Default</title>
</head>
<body>
   <asp:DataGrid runat="server" id="NamesControl"
                 AutoGenerateColumns=true>
     <HeaderStyle BackColor="#00aaaa">
     </HeaderStyle>
   </asp:DataGrid>
</body>
</html>
```

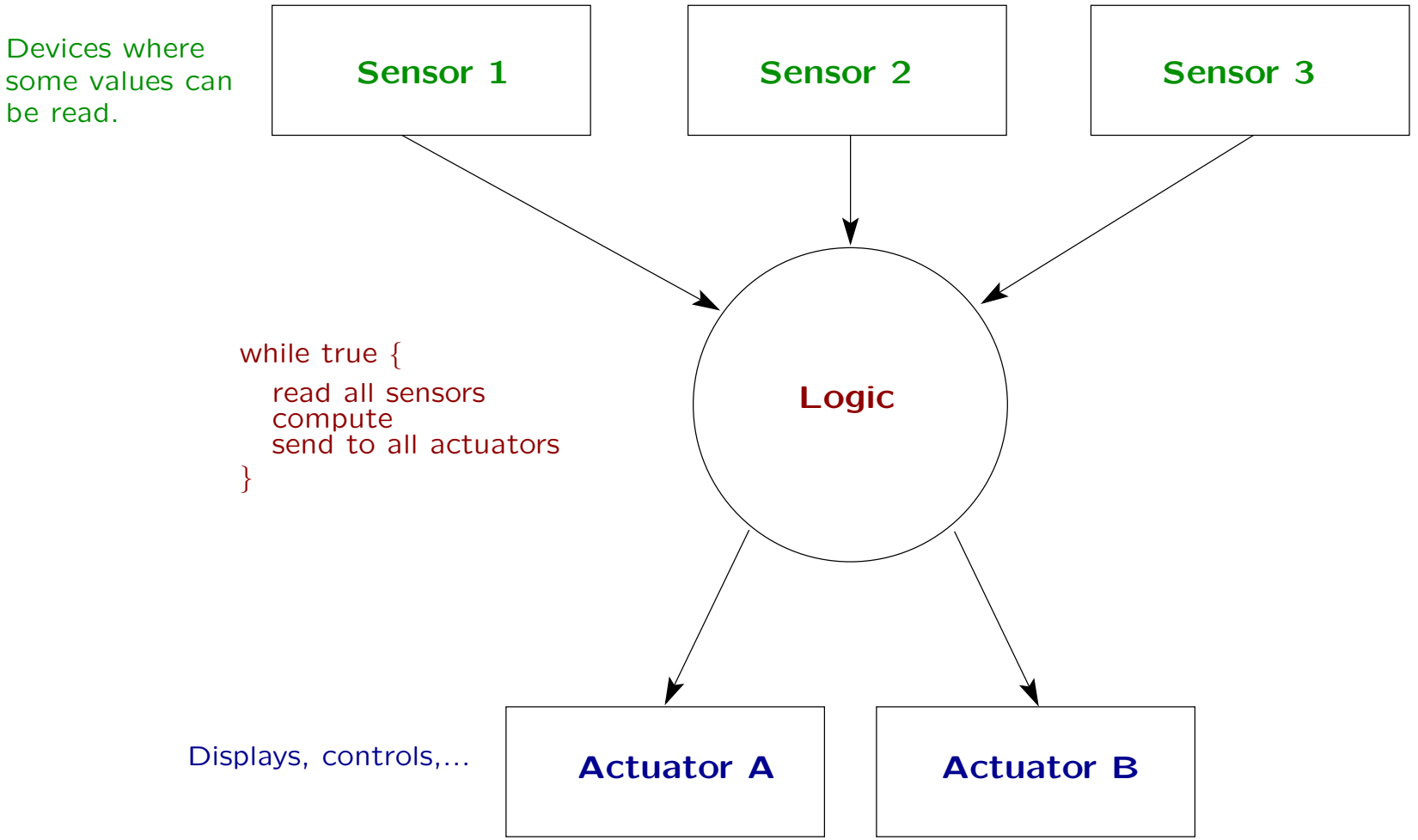
Sample .CS code

```
using ...;
using Npgsql;

namespace TestWeb
{
  public partial class Default : System.Web.UI.Page
  {
    private void Page_Load(Object sender, EventArgs e)
    {
      DataSet ds = new DataSet();
      NpgsqlDataAdapter da = new NpgsqlDataAdapter(
        "SELECT * FROM hent_navne", Global.dbconn);
      da.Fill(ds);

      NamesControl.DataSource = ds.Tables[0];
      NamesControl.DataBind();
    }
  }
}
```

Model-View-Controller



Sensor-Compute-Control



Styles

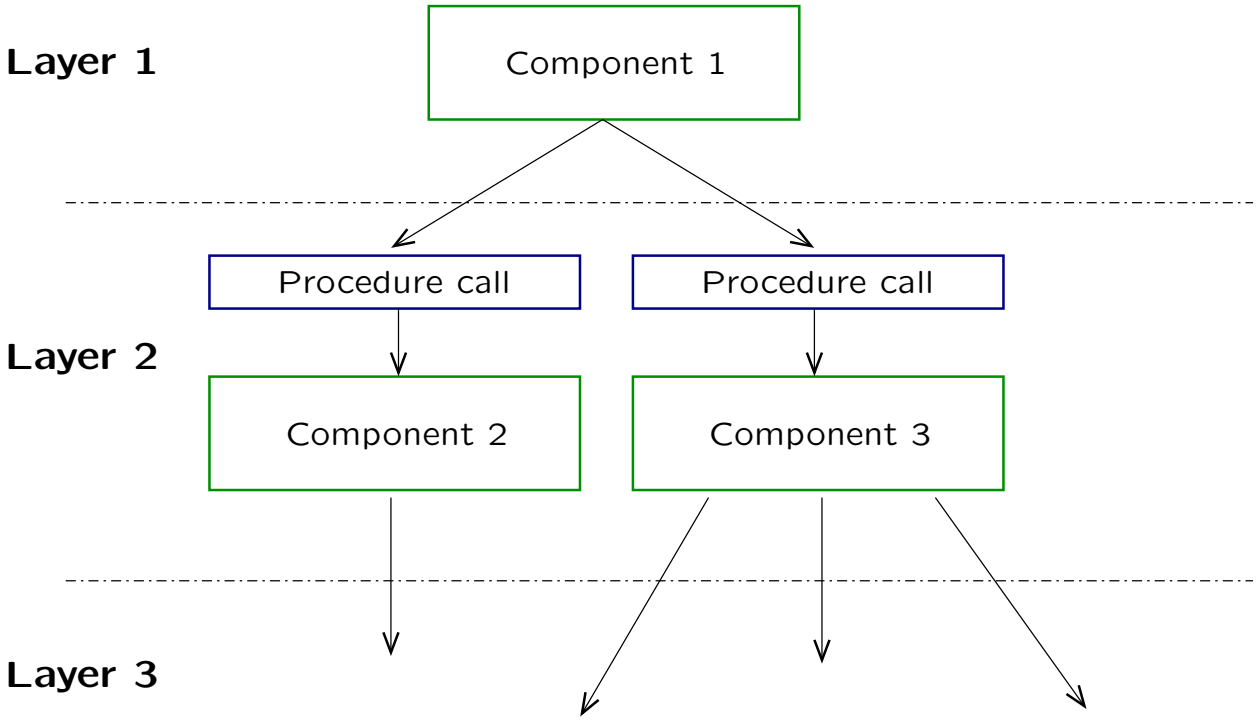
Definition: An architectural style is a named collection of architectural design decisions that

- are applicable in a given development context,
- constrain architectural design decisions that are specific to a particular system within that context, and
- elict beneficial qualities in each resulting system.

You know at least two:

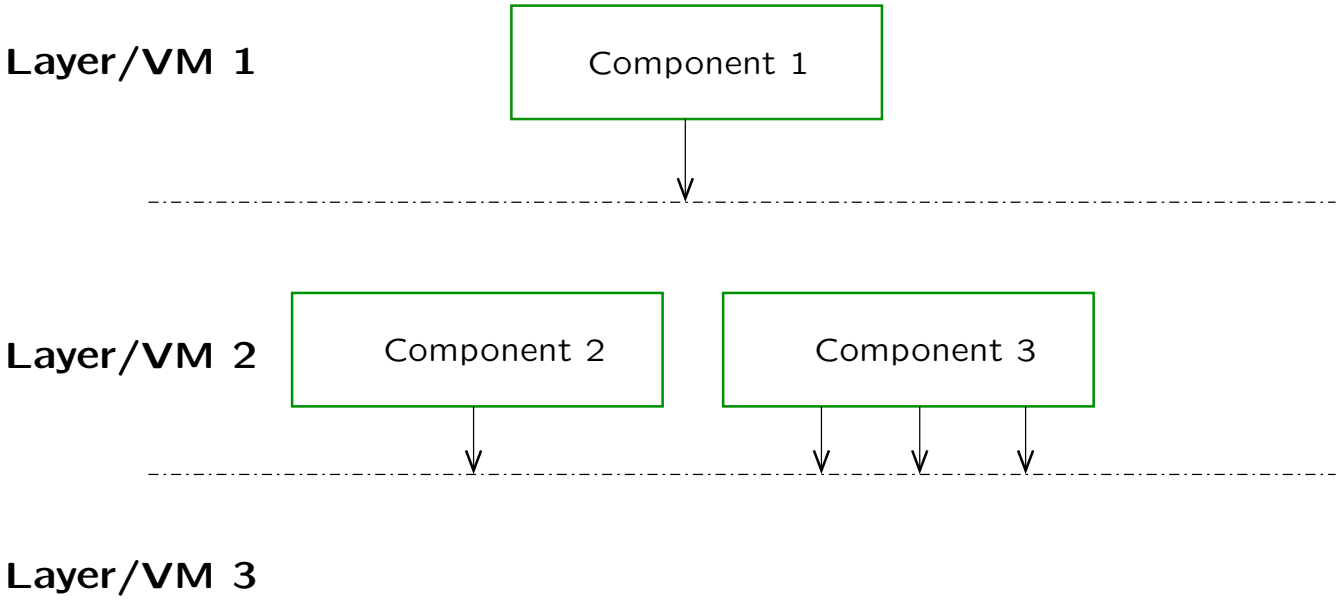
- Structured programming (functions, sequence, loops),
- Object-oriented programming.

Layered style



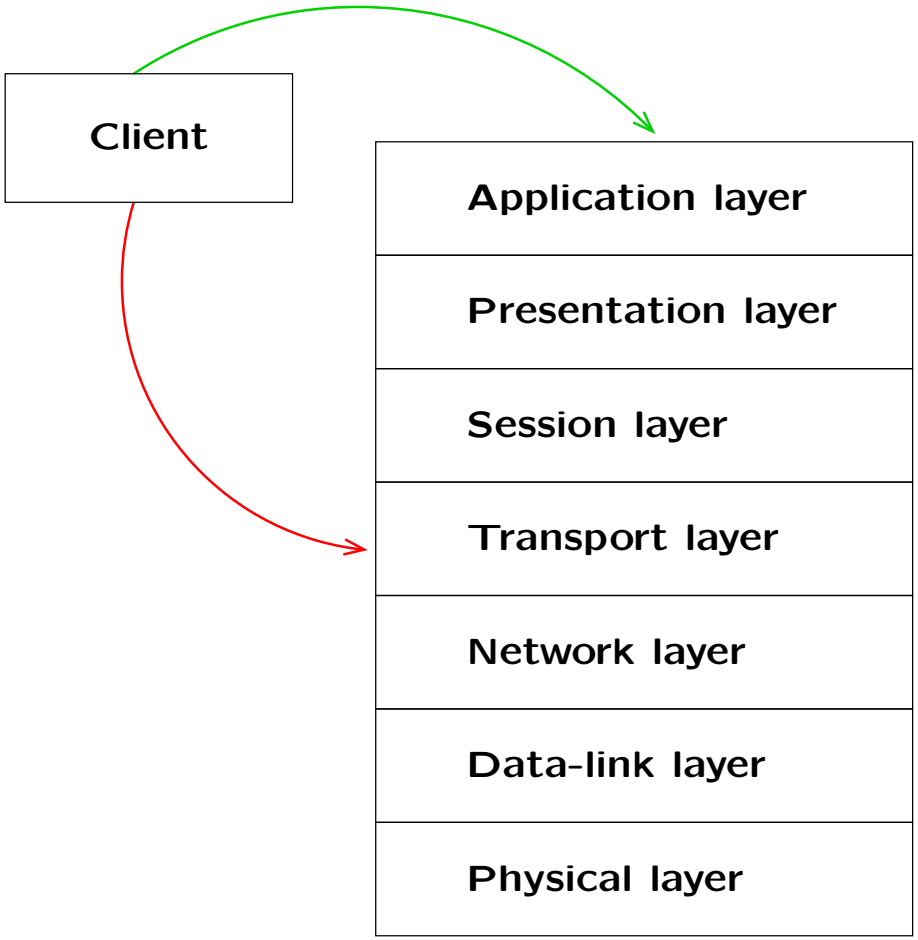
Virtual machine

Each layer is a virtual machine. Each layer offers a set of services, that may be accessed by a component located in the layer above.



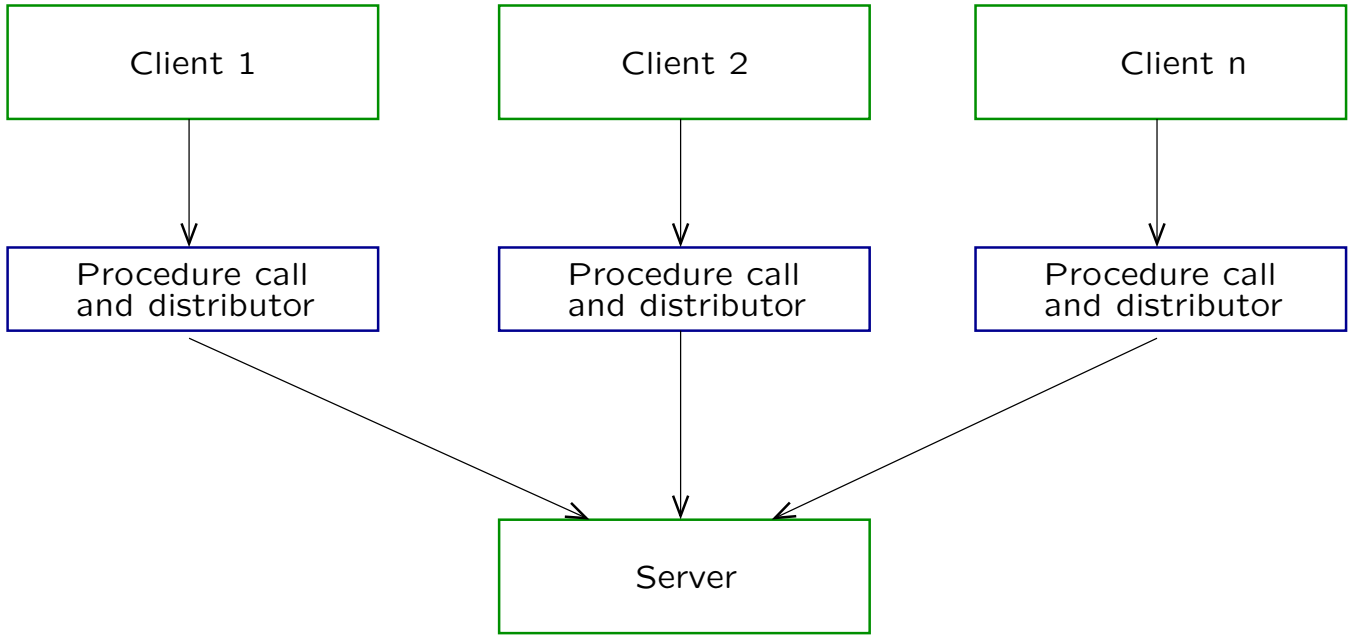
Cautions: Strict virtual machines with many levels can be relatively inefficient.

Example: The OSI model



Client-server

Clients send service requests to the server. Communication is instantiated by the clients.

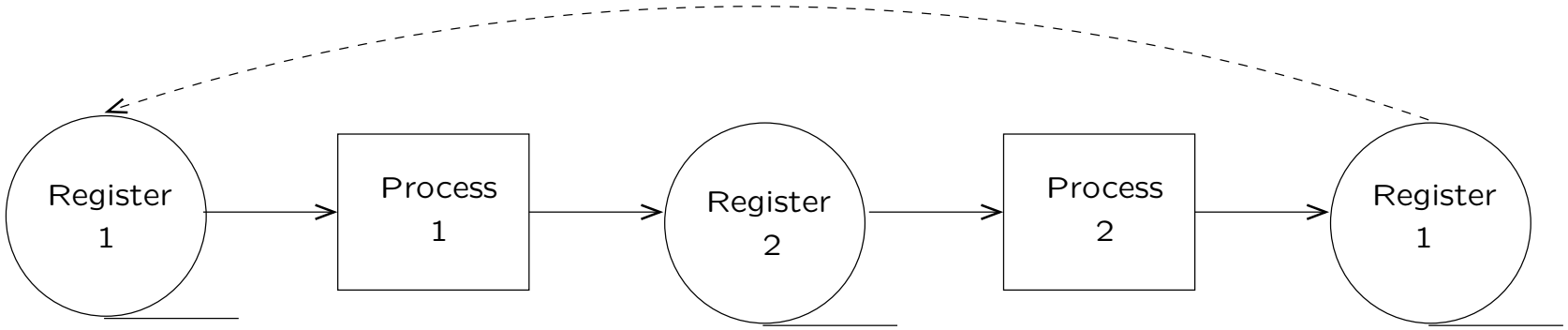


Cautions: Single-point-of-failure, network bandwich, number of requests.

Dataflow Styles

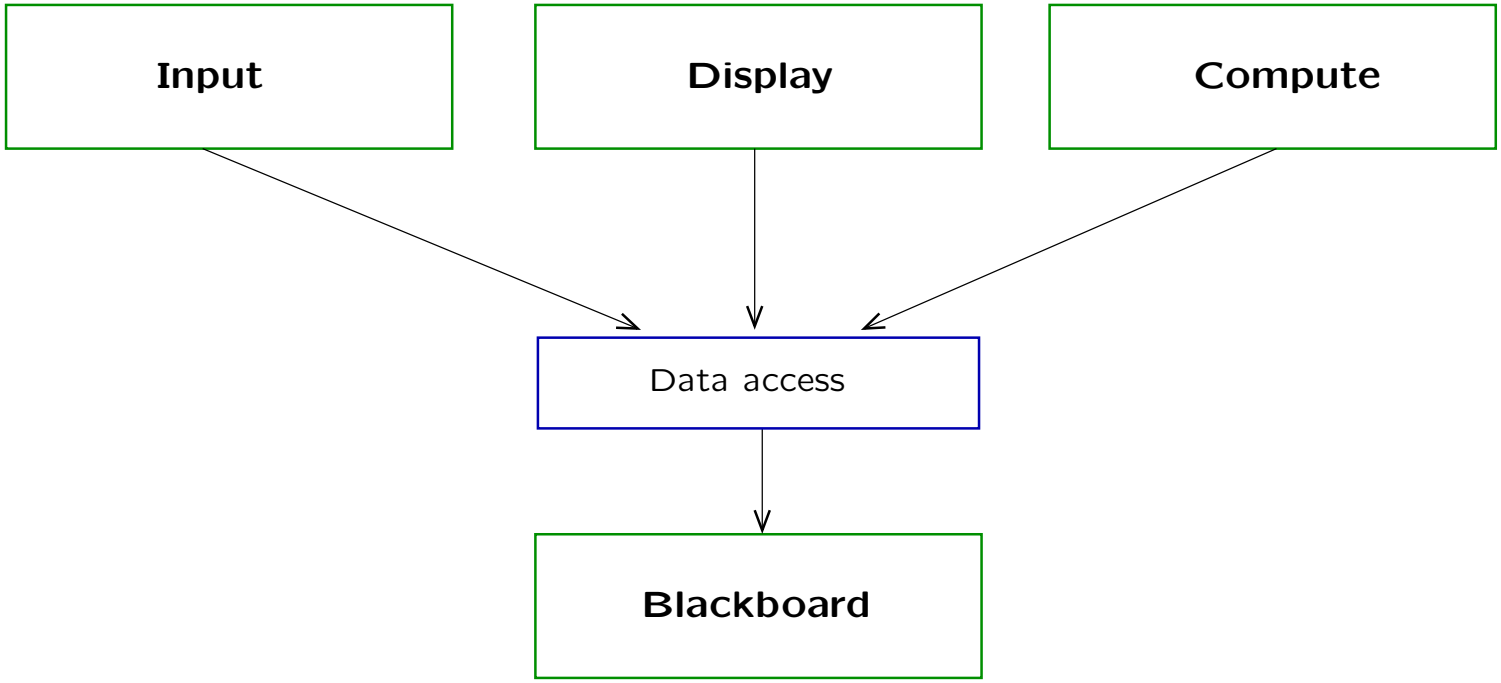
Point of view: Movement and processing of data. Think of the UNIX command interpreter example (denoted pipe-and-filter).

Batch-sequential:



Shared state

Blackboard: Independent program access and communicate exclusively through a global data repository.

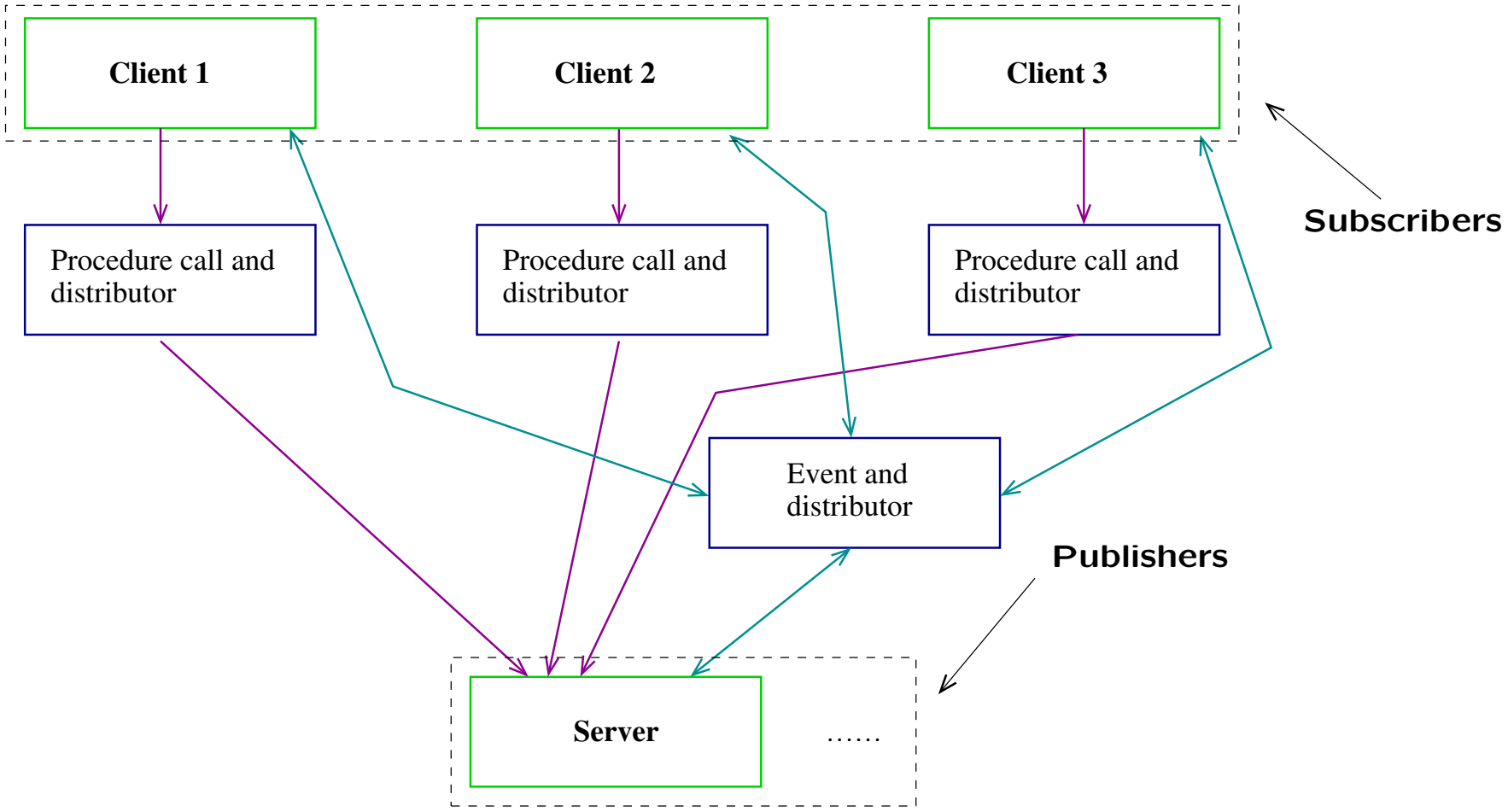


Implicit invocation

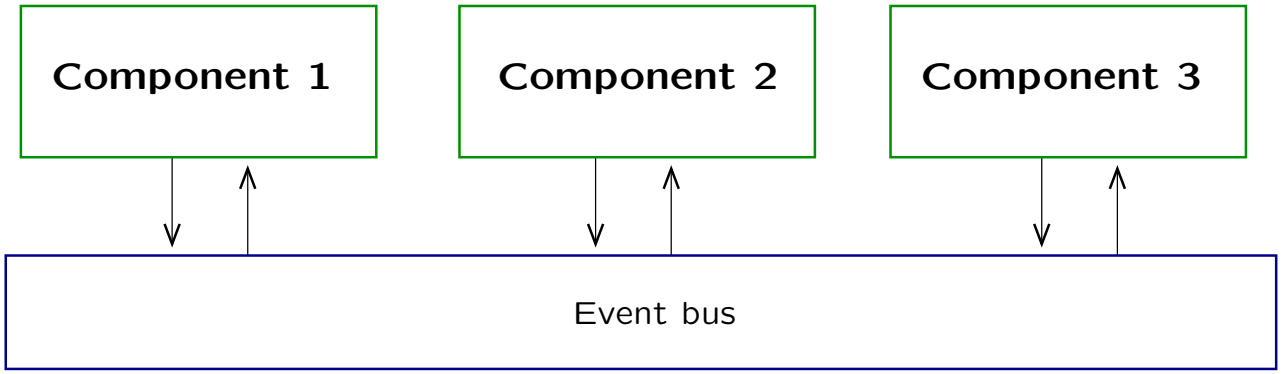
Publish-subscribe: Subscribers register/deregister to receive specific messages or content. Publishers maintain a subscription list and broadcast messages to subscribers.

Event-based: Clients communicate using the event bus, not directly to eachother.

Publish-subscribe

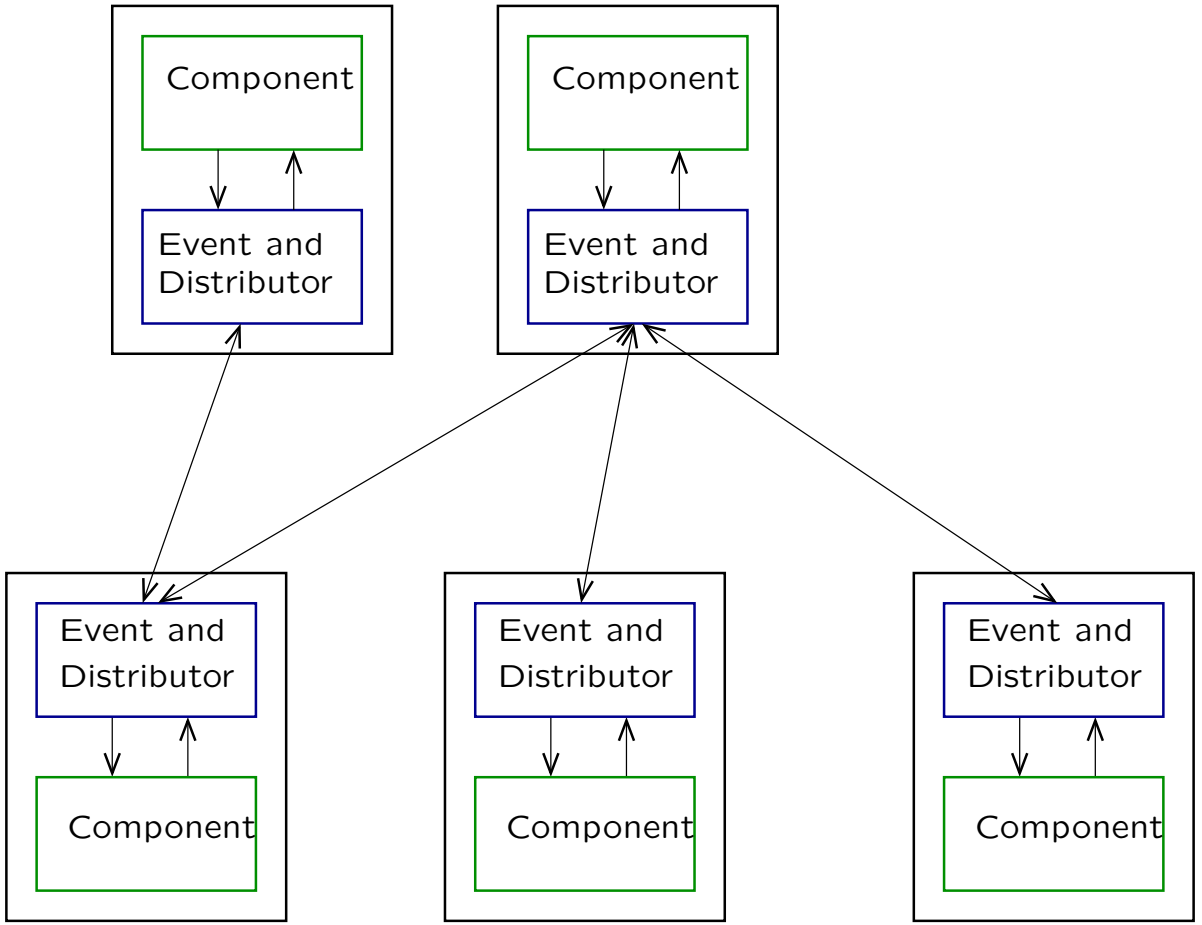


Event based



Cautions: The order of the messages may not be preserved.

Peer-to-Peer


Component
Event and Distributor
Component
Event and Distributor
Event and Distributor
Component
Event and Distributor
Component
Event and Distributor
Component

Distributors

Some examples:

- CORBA,
- XMLRPC,
- SOAP (Simple Object Access Protocol),
- Programming language specific (e.g. Pyro in Python).
- Your own protocol.

Nameservices

- DNS
- P2P-networks: Bittorrent, Gnutella, etc.

SOAP example

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

Source: SOAP on Wikipedia

Example using VB.NET

Can easily be used in .NET language (C#, VB.NET) etc. using a WSDL (Web-service description language) file.

```
Sub Button1_click()
Dim objSOAPClient As Object
objSOAPClient = CreateObject("MSSOAP.SoapClient")
objSOAPClient.mssoapinit(http://ServerName/VirtualDirectory/wsdlfile.WSDL)
MsgBox(objSOAPClient.Add(100, 400))
End Sub
```

Source: http://www.c-sharpcorner.com/UploadFile/mbmehta/soapIntroduction11112200507281 8AM/soapIntroduction1.aspx

Your own protocol

Consider an Internet e-mail server allowing other services to decide whether a mail should be rejected or accepted (useful for spam checking).

The mailserver postfix has such a support by a simple plaintext protocol. Postfixs connects to a service by TCP at localhost and sends the following information about the mail.

```
request=smtpd_access_policy
protocol_state=RCPT
protocol_name=SMTP
helo_name=some.domain.tld
queue_id=8045F2AB23
sender=foo@bar.tld
recipient=bar@foo.tld
recipient_count=0
```

```
client_address=1.2.3.4
client_name=another.domain.tld
reverse_client_name=another.domain.tld
instance=123.456.7
```

The service should answer:

- `DEFER_IF_PERMIT`: Service temporarily unavailable. The sending server should try again.
- `DUNNO` or `OK`: Accept.
- `REJECT`.

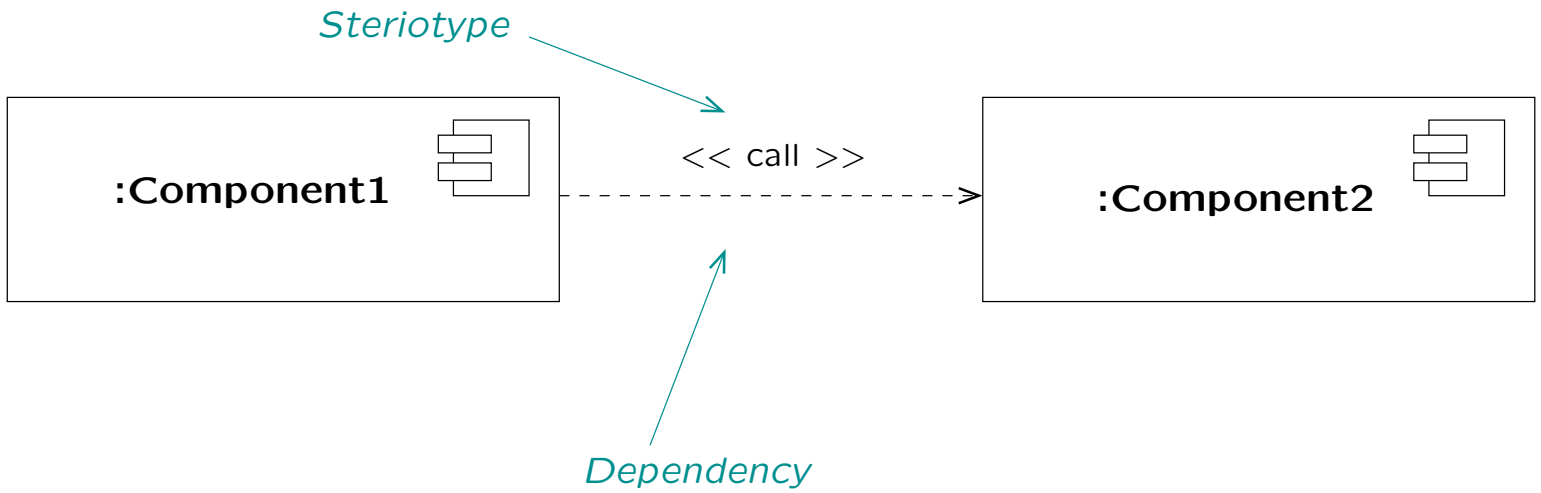
Modelling

Definitions: An architectural model is an artifact that captures some or all of the design dicisions that comprise a system's architecture. Architectural modelling is the reification and documentation of those design decisions.

Methods for documentation:

- Natural language,
- UML,
- Architecture Description Language (ADL).

UML



ADL

Architecture description language is

- A language for specifying a software system's architecture,
- Provide support for: Components, Connectors, Interfaces, and Configurations.

We will discuss Darwin.

Darwin

The Darwin language is defined by the following regular expression:

(**component** { (

- **provide** *identifier* ; |
- **require** *identifier* ; |
- **inst** (*identifier*: *component* ;)* |
- **bind** (*component*.*identifier* -- *component*.*identifier*)*

)* })*

Example

Let us consider the CPH STL again. At the top level we have

- Containers,
- Iterators,
- Realizators,
- and Storage policies.

We want to describe the architecture using Darwin.

```
component AVL_Node {
  provide StoragePolicy;
}

component AVL_Balancer {
  provide Balancer;
}

component SearchTreeFramework {
  require StoragePolicy;
  require Balancer;
}
```

```
component Set {
inst
  R: SetRealizator;
  B: AVL_Balancer;
  N: AVL_Node;
bind
  R.StoragePolicy -- N.StoragePolicy;
  R.Balancer -- B.Balancer;
}
```

Final words

- The book is great, buy it if you're interested in the topic.
- Courses on architecture are available at ITU and SDU.
- A relavant topic for a Master's thesis/project.