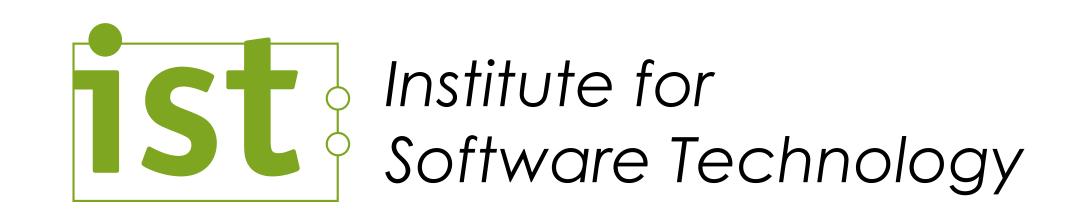


#### 1. Introduction

Engineering Web and Data-intensive Systems



#### Introduction

- Content Overview
- Relation to Software Engineering
- Types of Web Systems
- Contribution to Data-intensive Systems



## Chapter I. Introduction

#### Resources

- G. Kappel, B. Pröll, S. Reich, and W. Retschitzegger (editors):
   Web Engineering: The Discipline of Systematic Development of Web Applications.
   John Wiley & Sons, Chichester, England, 2006.
- M. Fowler et al.:

   Patterns of Enterprise Application Architecture.

   Addison-Wesley, 2006
- G. Rossi, O. Pastor, D. Schwabe, and L. Olsina (editors):
   Web Engineering Modeling and Implementing Web Applications
   Springer, 2008
- M. Seidl, M. Scholz, C. Huemer, and G. Kappel:
   UML @ Classroom An Introduction to Object-Oriented Modeling
   Springer, 2015
- ... and many, many online resources and tutorials for self-study

# Engineering Web (Systems) And Data-Intensive Systems

5

## Engineering

### Engineering

- Computer Science is an engineering discipline
- Engineering approaches consist of five steps
  - A. Gain background knowledge
  - B. Understand and precisely formulate the problem
  - C. Design a solution
  - D. Realize and implement that solution
  - E. Verify the solution

- Software Engineering as a part of Computer Science researches, creates, and applies methods and tools to conduct those engineering steps to create software systems that solve problems.
- In software (and data) engineering, models are among the central artifacts in steps B to D.
- Hence, we'll deal with modeling in greater detail.

## Software Engineering

Software Engineering is the part of Computer Science that deals with construction, usage, and enhancement of (usually big) software systems.

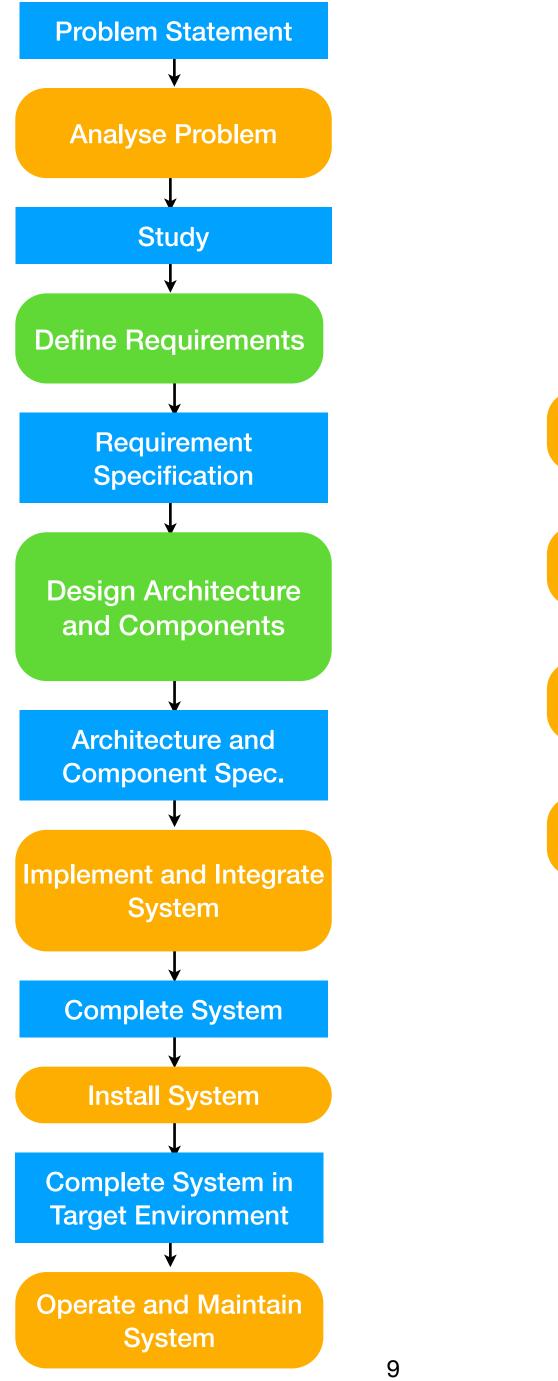
• A **Software System** is a set of programs together with accompanying artifacts that are necessary or helpful to use and operate the system.

 Among those artifacts are requirements specifications, designs, functional specifications, test suites, user manuals, configuration files, scripts, databases, deployment diagrams, release plans, ...

8

#### Software Lifecycle

This does not imply a sequential "waterfall" process model!



Continuous Activities

Manage Project

Control and **Document Process** 

Manage Configurations

**Assure Quality** 

Legend

Activity

**Artifact** 

### Software Quality

A software system is expected to be...

- correct
- reliable
- robust
- efficient
- extensible
- configurable
- scalable
- maintainable

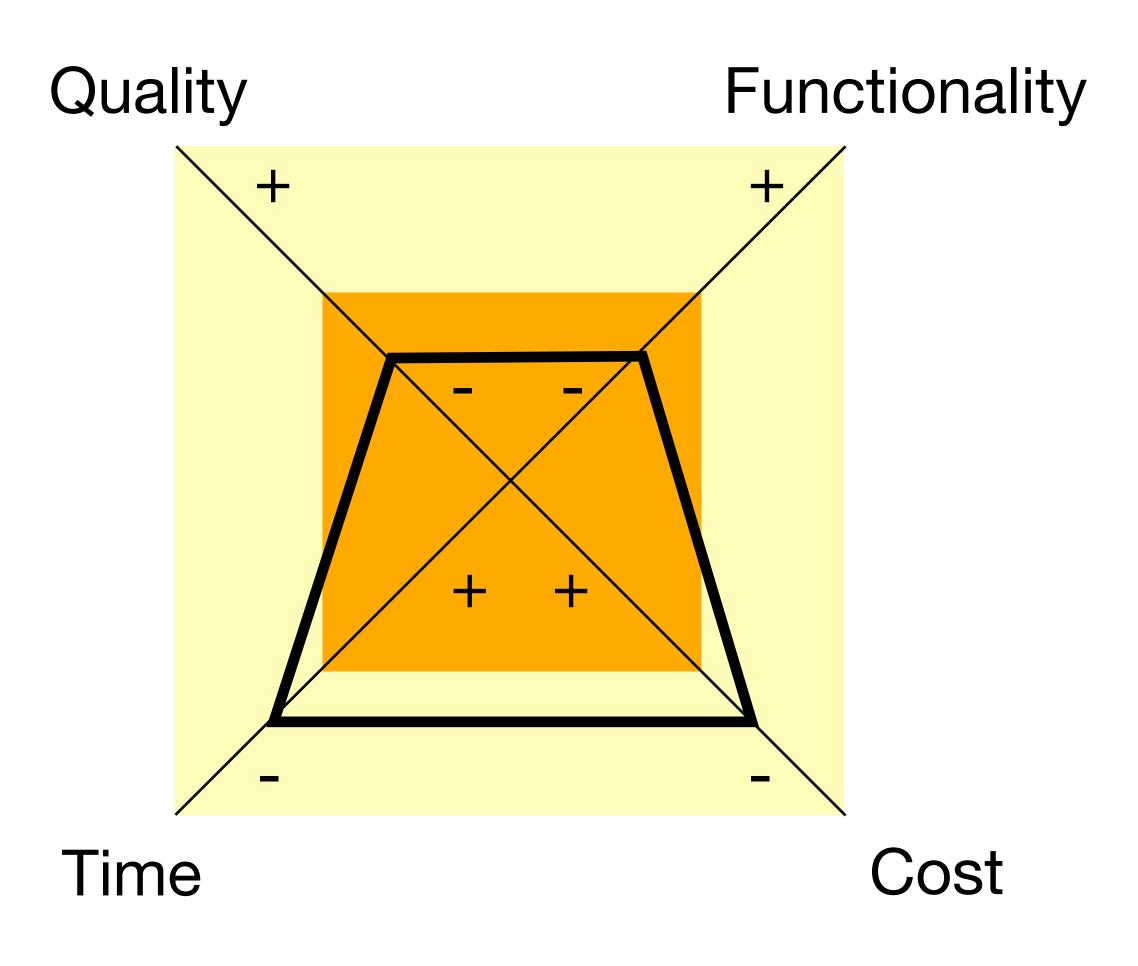
reusable

easy to use

- up-to-date
- safe
- secure
- well documented
- •

We'll study aspects of software quality that are specific to web and data-intensive systems.

## "Devils Square" (according to Harry Sneed)



- Goals depend on each other
- Orange area represents productivity of a project, assumed to be constant
- Examples:
  - Limiting cost and time reduces functionality and/or quality (black trapezoid)
  - Increasing functionality implies higher cost, more time, or lower quality
  - Quality doesn't come for free. Higher quality means higher cost and more time for the same functionality

#### Security

- With Web and Data-intensive Systems, we face specific challenges:
- 24/7 online
- distributed, ubiquitous
- massive attacks conducted
- high security expectations
- strict regulations (e.g., GDPR)

- high risk to compromise
  - the system itself
  - the data stored on the system
  - the reputation and wealth of the operating company

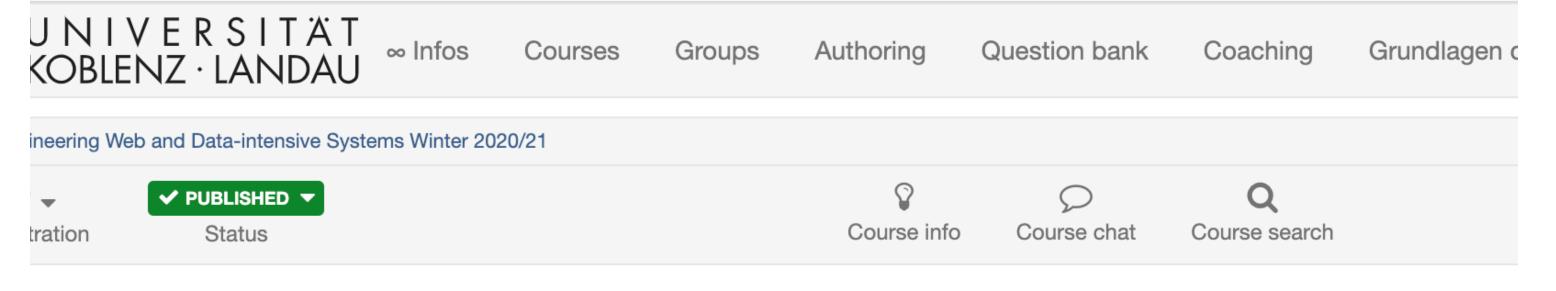
## Web Systems

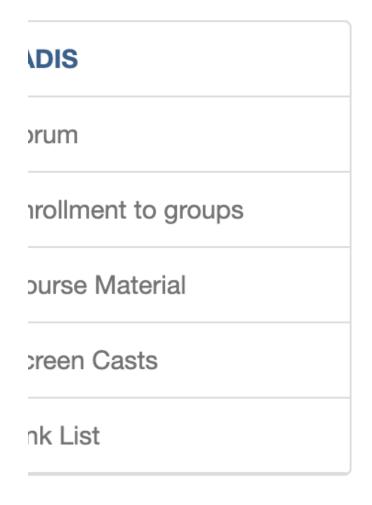
## Web Systems (by example)

#### Definition

A **Web Application** is a software system based on technologies and standards of the World Wide Web that provides Web specific resources such as content and services through a user interface, the Web browser.

[cf. Kappel 2006 WE]







#### Engineering Web and Data-intensive Syst

▼ Hide description

Access to course material, examples, and discussion forum.

#### **♀** Forum

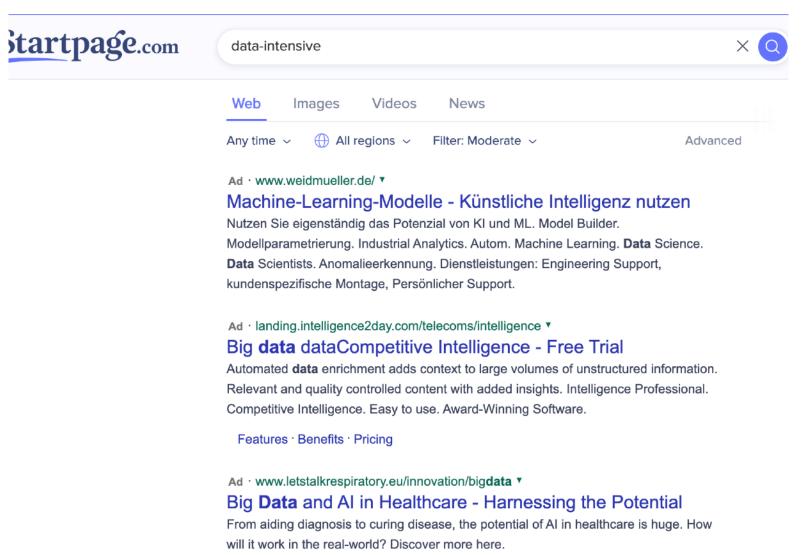
**EWADIS Discussions and Announcements** 

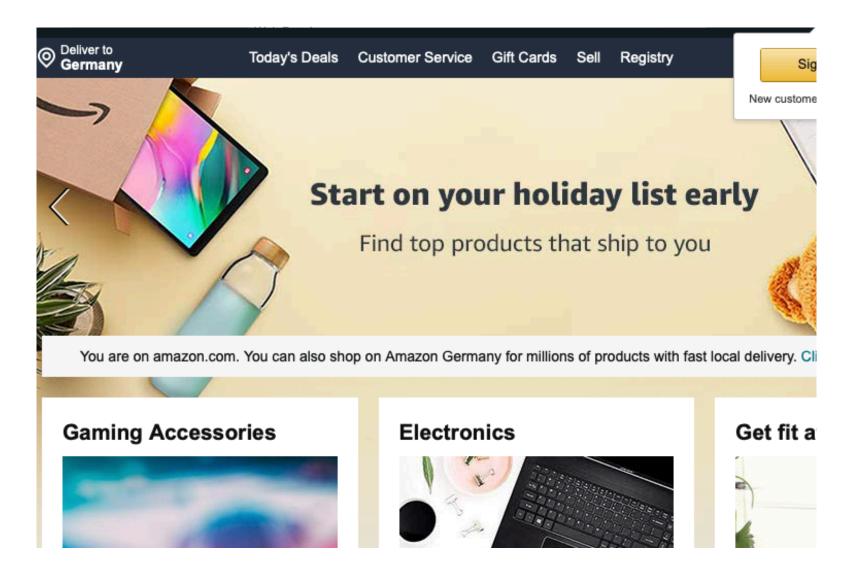
#### **◆** Enrollment to groups

Course Material

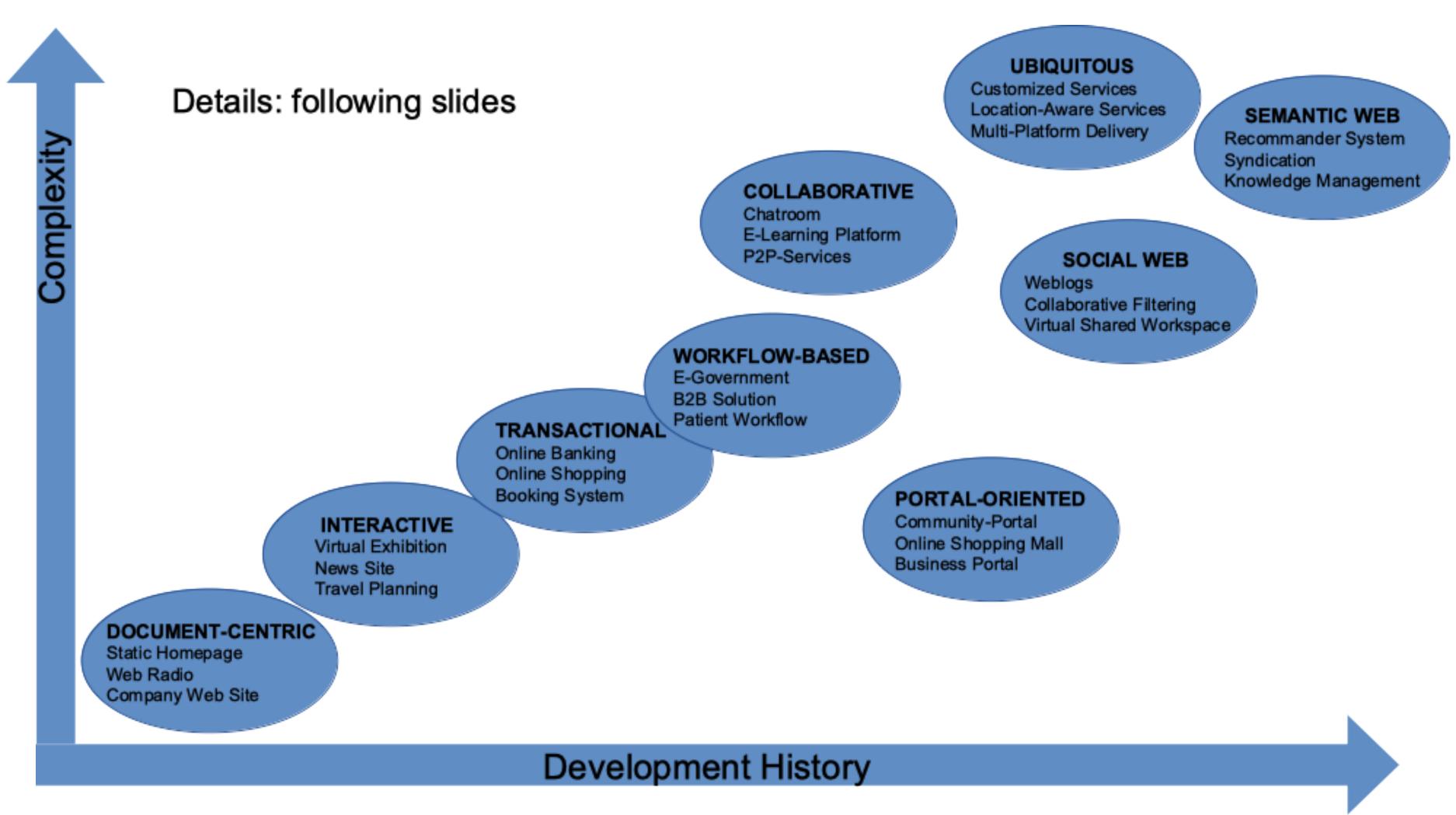
Screen Casts

Link List





#### Spectrum of Web Applications



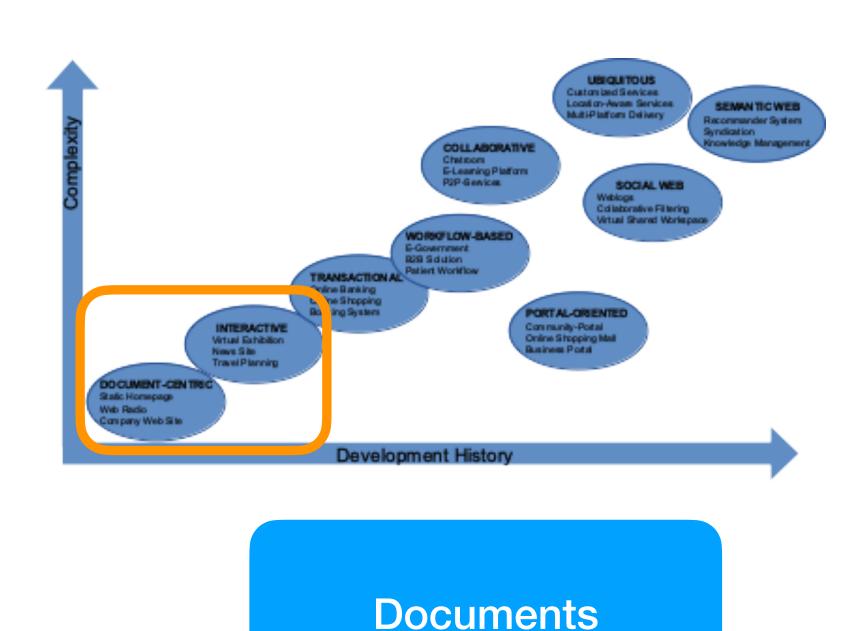
- --- Birth of The Web in 1989 <a href="http://info.cern.ch">http://info.cern.ch</a>
- ---> https://home.cern/science/computing/birth-web

## Categories I

- Document Centric Web Sites

   using HTML,
   static documents,
   updated manually

   Examples
   static homepages
- Interactive Web Applications
   ... using CGI, HTML forms, with pages
   generated automatically in response to user
   input (read-only)
   Examples
   news sites, timetable information

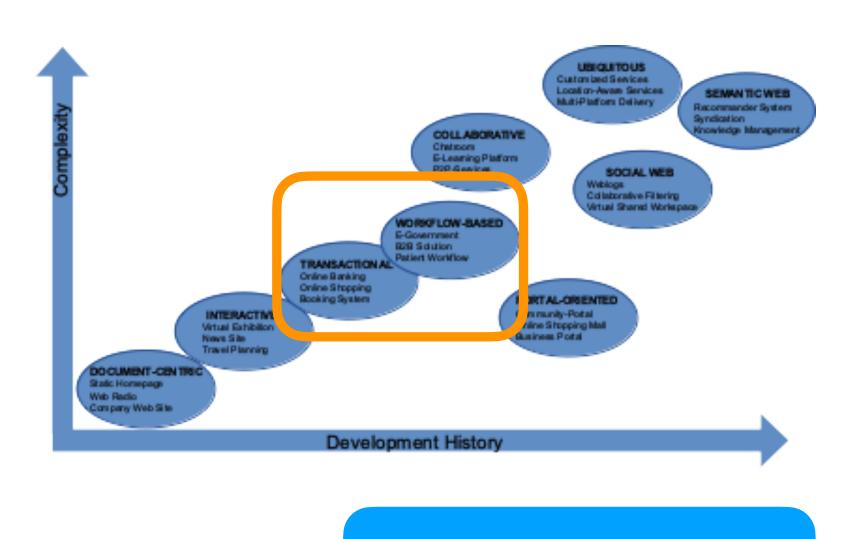


Interaction

### Categories II

- Transactional Web Applications ... allow user to update underlying content, i.e. data base (read-write) **Examples** conference registration, room booking
- Workflow-Based Web Applications ... using Web Services, complex interactions involving several partners (users, companies, public authorities), in a structured collaboration

**Examples** e-commerce, e-government



**Databases** 

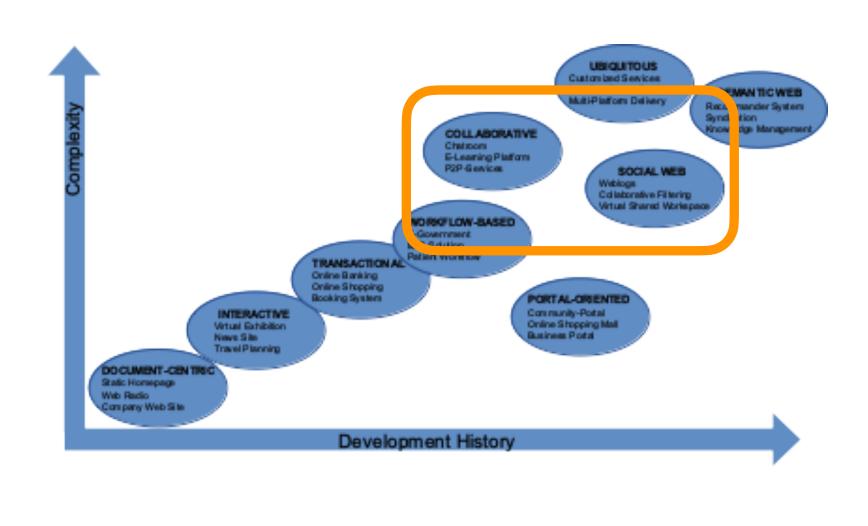
Business **Processes** 

18

### Categories III

Collaborative Web Applications
 ... intense cooperation of many
 users in collaborative work,
 generate/edit/manage shared
 information
 Examples
 groupware, wiki, bscw, ...

## SocialWeb ...collaboration as part of non-anonymous personal cooperation Examples weblogs, Facebook, ...



Synchronization

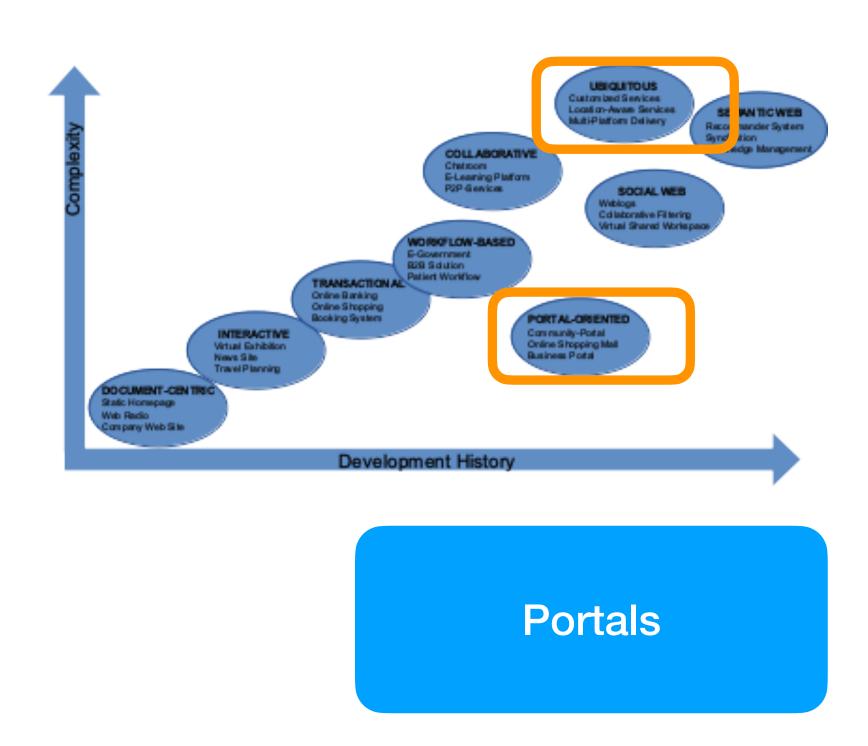
Social

#### Categories IV

- Portal-Oriented Web Applications

   supply information from and interact with heterogeneous sources; single point of access to handle separate, potentially heterogeneous sources of information Examples search engines, marketplace portals
- Ubiquitous Web Applications

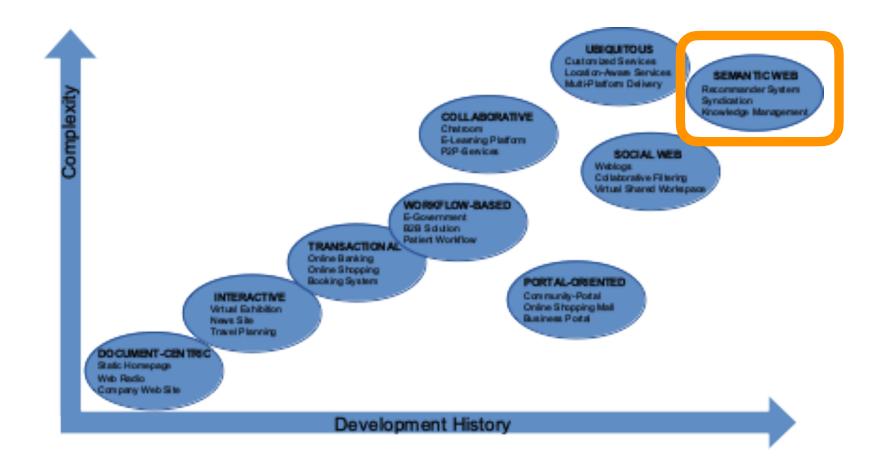
   customized services anywhere for any device, personalized, location aware, multi-platform
   Examples device-independent apps



Ubiquitous

### Categories V

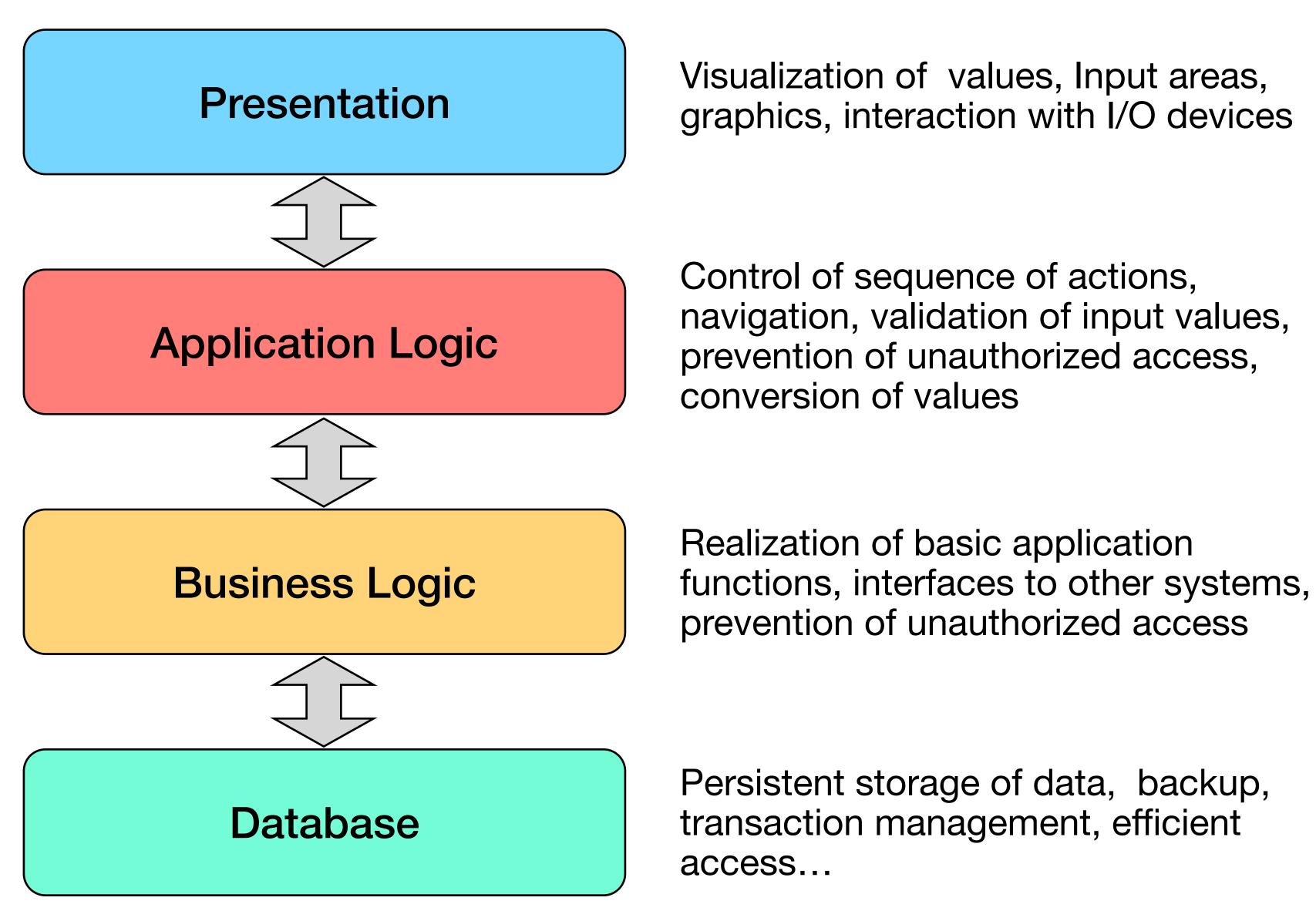
Semantic Web...
 offer information not only
 for humans, but in a
 machine-readable form,
 enabling automatic knowledge
 management
 Examples
 ontology-based knowledge sources



(formal)
Semantics

## High Level Architecture of (Web-) Applications

#### 4 Layer Architecture



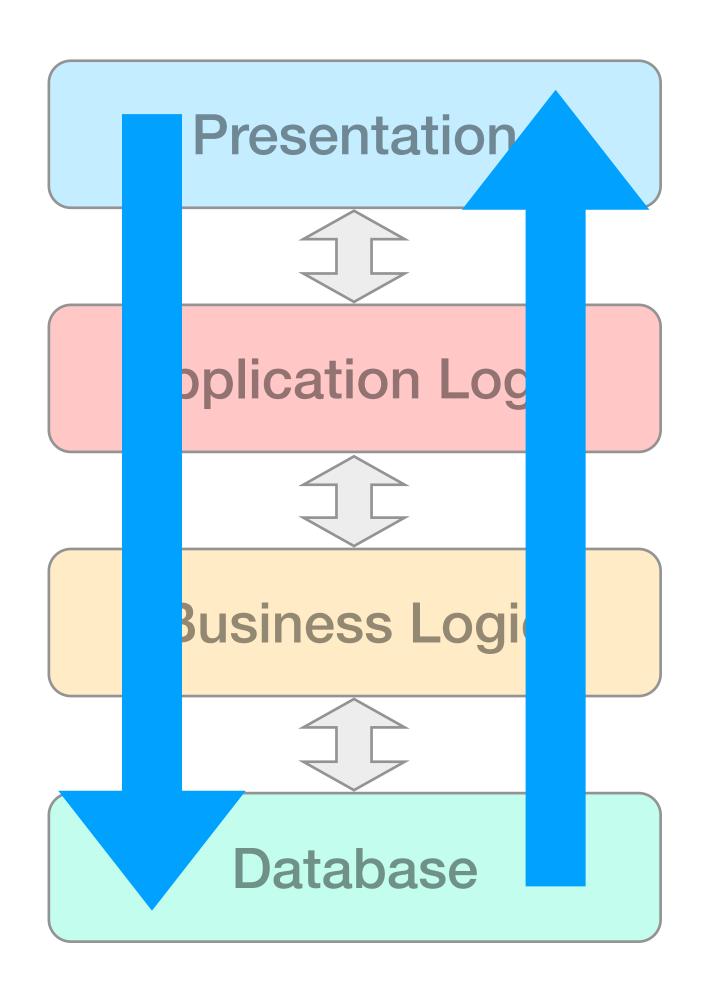
#### Request Processing

User requests a resource or triggers a function

Evaluate request and parameters, select business functionality

Execute business functions possibly using partner systems

Fetch or modification of persistent data



Encode and transform into presentation specific representations, render the results

Translate into appropriate formats, change state, select appropriate response

Prepare, compute, aggregate, filter results, convert into suitable units

Retrieve result data

## Data-intensive Systems

#### Data-intensive Systems

- Usually means...
  - high complexity of data structures
  - and/or big amounts of data
  - and/or high computing costs to get results

- Usually realized with...
  - distributed data storage
  - distributed computing
  - specialized computing frameworks to balance computing costs

#### EWADIS deals with

- different logical (implementation) data models: relational, document/keyvalue, and graph
- foundations of data distribution and reliability issues
- (the other topics are key content of the Big Data course)

#### Data-intensive Systems

- Data is a representation of information
- Data needs to be...
  - stored,
  - queried,
  - protected,
  - interpreted,
  - transformed,
  - aggregated,
  - transferred,
  - managed,
  - archived,
  - deleted,
  - •

- Different data content and different access schemes require:
  - specific data models,
  - storage formats,
  - (query) languages,
  - algorithms,
  - transfer protocols,
  - •

We'll study some types of databases: relational, document-centric, key-value based, and graph based.

#### Outlook

- Introduction
  - Overview
  - Quality Goals
  - Requirements
- Architecture of Web Applications
  - Styles, Patterns, Templates
- Modeling
  - Structural and behavioral models
  - Web-specific modeling
  - Transformation

#### Communication

- Networks, Protocols, Encryption
- Server side and client side components
- Data Representation
- Persistence
  - Data Models
  - Relational, Document, and Graph Databases
  - Distributed Data
- Security
  - Attacks and Problems
  - Countermeasures
  - Access Control

#### What we have learned...

#### Introduction

- √ Content Overview
- √ Relation to Software Engineering
- √ Types of Web Systems
- √ Contribution to Data-intensive Systems

