Software Engineering CSC 648-848 Fall 2024

09-17-24

Software Architectures and Design Patterns

Multimedia and Search Architectures Design Patterns

Prof. D. Petkovic

Objective

- Understand importance and role of SW architecture design at high level (generally agnostics to specific implementation)
- Arch. Design patterns: usage and benefits
- Examples
- UML – Unified Modeling Language
- (NEW: GenAI Can help a bit too)

- Multimedia and Search architectures (data and metadata organization, UI)

Some Analogy

- Architecture of the house vs. architecture of SW system
- House:
  - Room arrangement, access, proximity
  - Flow of people/material
- SW:
  - Components arrangement, interfaces, connectivity
  - Flow of data/actions

Architectural design

“Process of designing SW system organization”

- Driven by non-functional specs (which are usually driven by business needs and cost)
- Facilitates discussion about the SW design (needs to be *reviewed and adopted*)
- Needs to be followed and enforced (or revised)
- Helps in planning and cost estimates
- Helps communicate the design to the team and other stakeholders
- Always do high-level architecture review early in the process

Recall ACM slides on what makes great SW engineer – HOW TO MAKE DECISIONS is CRITICAL SKILL

Arch. Design and implementation involves many decisions

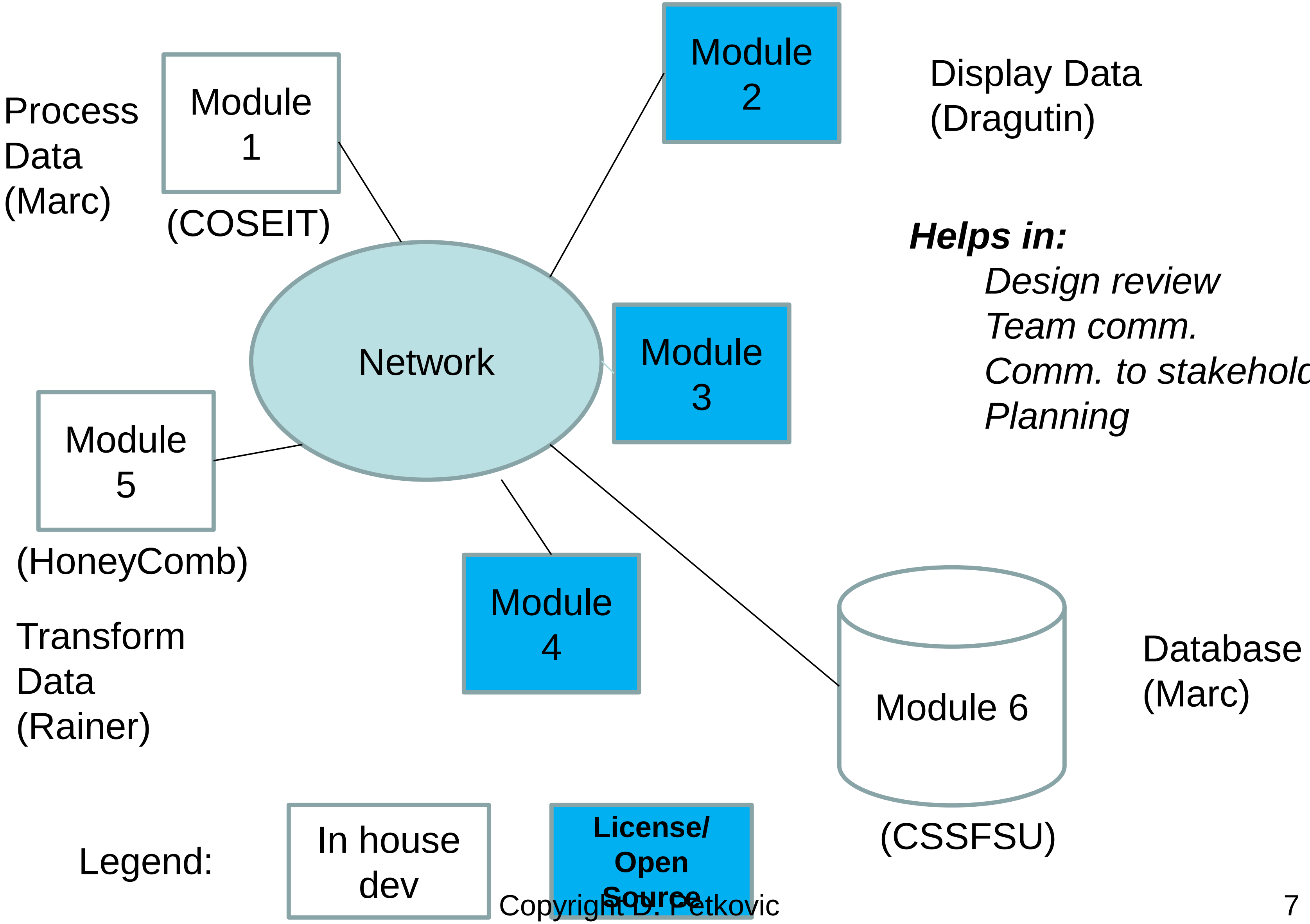
New GenAI tools can help in advising and suggesting – but YOU are in charge and repoinsible

So…ChatGPT can help too…as advisor

- Involve ChatGPT in architecture development cycle – as advisor

- https://dev.to/spy4x/how-chatgpt-can-help-design-system-architecture-for-your-applications-16dm
- https://chatgpt.com/g/g-iHXlDzolq-architecture-copilot
- https://www.siili.com/stories/chatgpt-for-strategic-software-architecture-decisions

Example: High level arch: show major systems, function, owners, location



Process Data (Marc)

Module 1

(COSEIT)

Module 2

Display Data (Dragutin)

Network

Module 3

Module 5

(HoneyComb)

Transform Data (Rainer)

Module 4

Module 6

Database (Marc)

(CSSFSU)

Legend:

In house dev

License/ Open Source

Helps in:

Design review

Team comm.

Comm. to stakeholders

Planning

Questions to ask in Arch. Design process (1)

- Can the application be mapped to known architectural solution or design pattern
- How to distribute the system across multiple processors
- Does the architecture support the system, data size and load requirements
- How to decompose system into subsystems – MODULAR design (high intermodule cohesion, loose coupling)
- Does the architecture satisfies non-functional requirements and costs
- How it should be reviewed and documented
- What existing farmeowrk can be used best to implement it

Questions to ask in Arch. Design process re implementation (2)

- Are the chosen tools and frameworks reliable, used in the marketplace, tested and supported
- Does the team know how to use them
- Licensing, legalese, cost
- Maintenance
- Stability of the provider
- Leverage ChatGPT and like as well but YOU are in charge and responsible

How to decompose system into subsystems – system design

- **This is crucial to good SW design irrespective of technologies used!**

- Partition and group requirements and define subsystems
- Specify subsystems – make it *modular*:
  - *Functionality* (well grouped and cohesive, focus on one category of functions)
  - *Interfaces* (ensure lose coupling, go for standard ones, well accepted in the market)

Goal of partitioning – modular design

- ***Modularity:***
  - *Intra-subsystem cohesion*: The subsystem has well defined and cohesive function (I.e. data gathering, not data gathering and data processing).
    - Test: can you describe its function in one simple sentence
  - *High intersystem independence or loose coupling* (I.e. separate data storage from data rendering)
    - Clean, well defined preferably standard interfaces
- Benefits:
  - Maintainability
  - Easy to debug, QA
  - Easy to optimize for security, speed

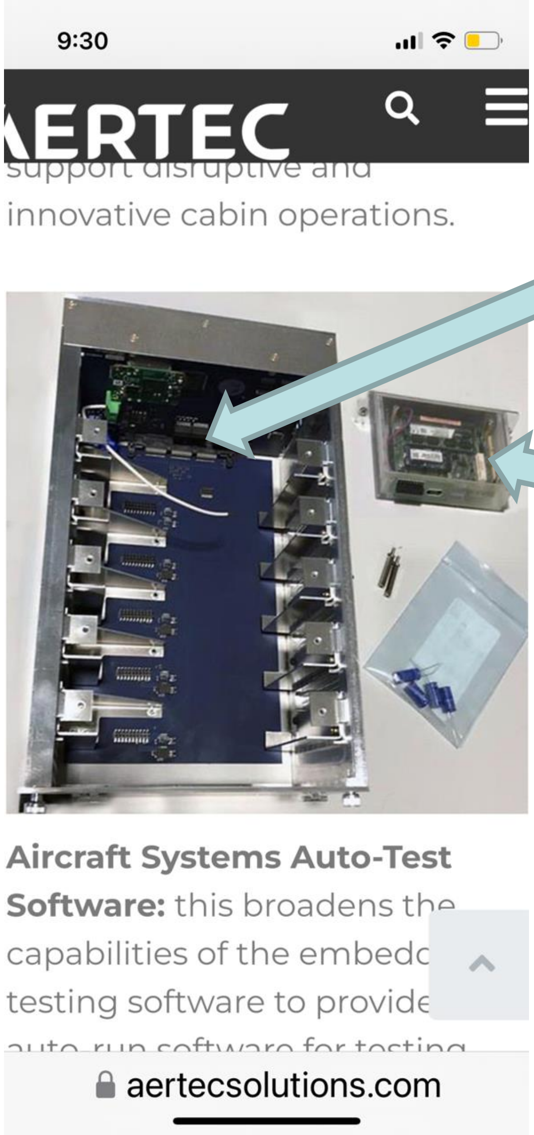
Important in the era of genAI – YOU do he Architecture, gen AI helps with smaller modules

- **"With modular programming, concerns are separated such that modules perform logically discrete functions, interacting through well-defined interfaces**." ref http://en.wikipedia.org/wiki/Modular_programming

Examples of partitioning

- System/method X has function to obtain *and* process data
  - Wrong! Make two separate subsystems/methods/classes. Why?
- Group all data access into one subsystem, one code. Test it well and enforce that only this code is used.
  - Why? Think of testing and tuning for performance or security
- Use industry standard APIs (application programming interface) and communication protocols
- Better use suboptimal but standard API and protocols than custom one of your own

Modularity standard and tried mandatory design pattern for electronics



9:30

AERTEC

support disruptive and innovative cabin operations.

Aircraft Systems Auto-Test Software: this broadens the capabilities of the embedc testing software to provide auto-run software for testing

aertecsolutions.com

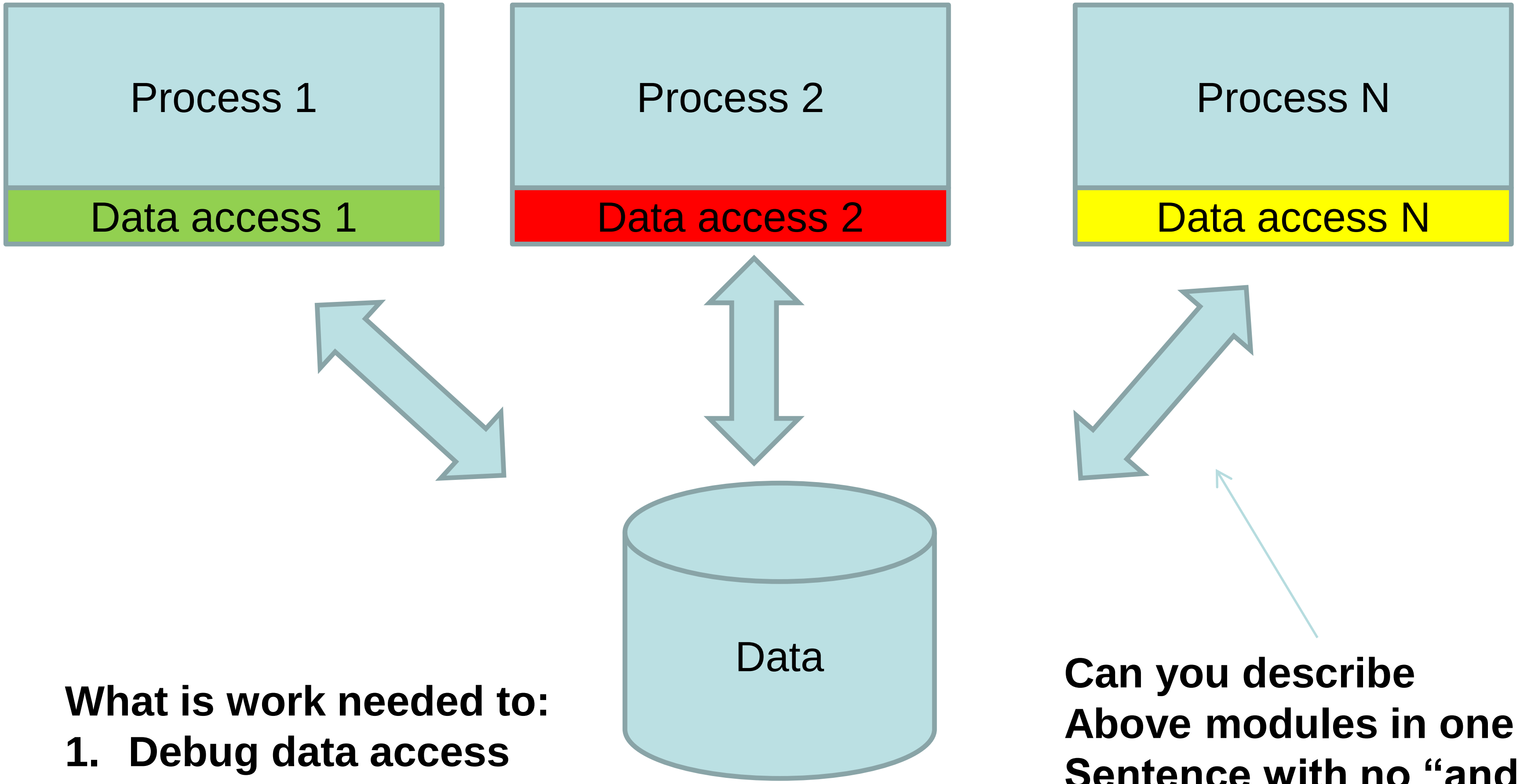
Your overall system Architecture with modular design and standard interfaces

Modules easy to replace: with well defined cohesive functionality and standard Interface (e.g. DB with SQL)

Modular design in action: Check Architecture solutions A and B in next 2 slides

- Ask yourself of the amount of work needed to
  - – Debug data access
  - – Maintain data access (e.g. data source API changes)
  - – Optimize data access for performance
- Do you prefer A or B architecture designs?

Architecture Solution A

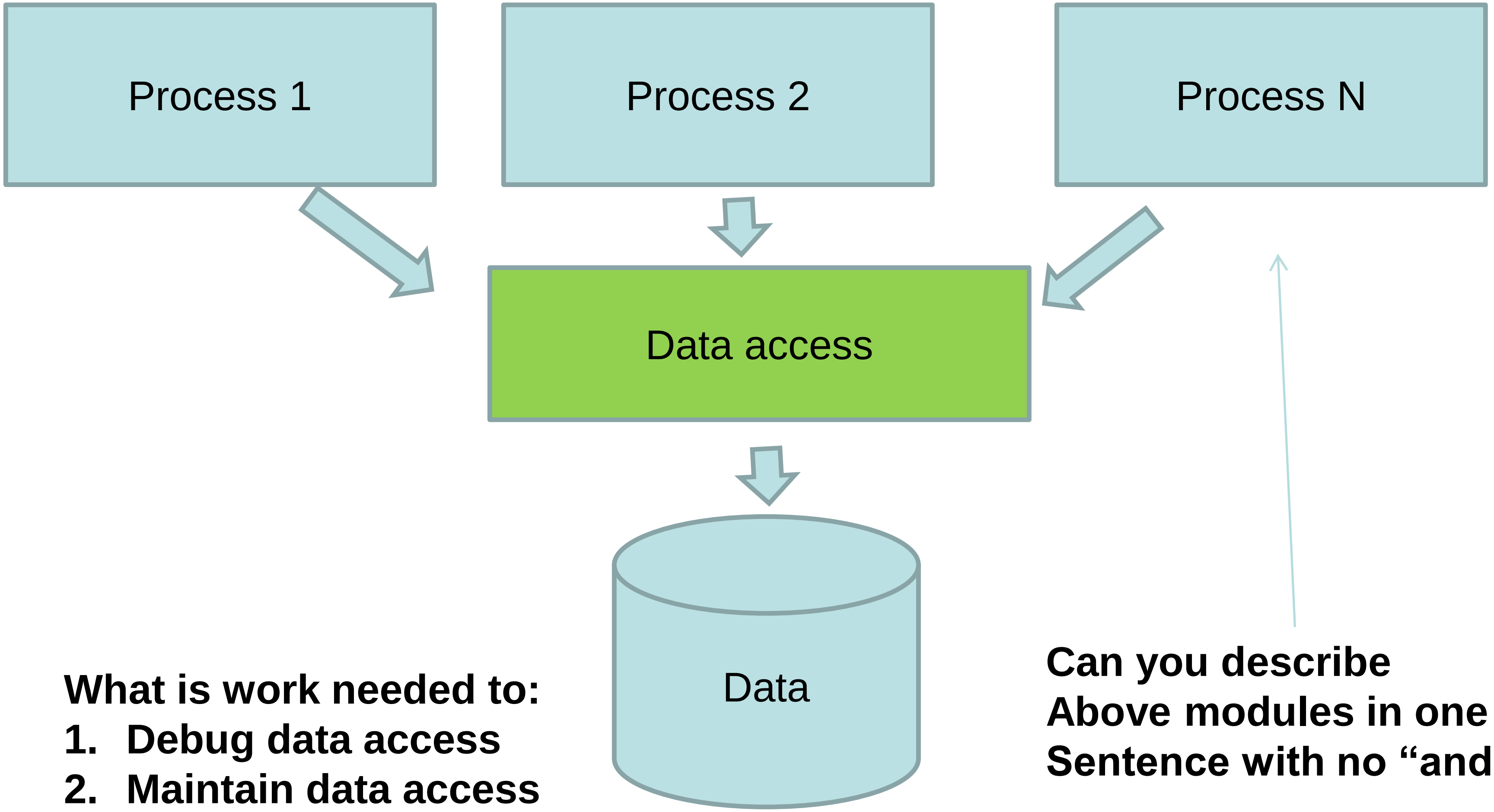


What is work needed to:

1. **Debug data access**
2. **Maintain data access**
3. **Optimize data access for performance**

Can you describe Above modules in one Sentence with no “and”?

Architecture Solution B



Can you describe
Above modules in one
Sentence with no “and”?

What is work needed to:

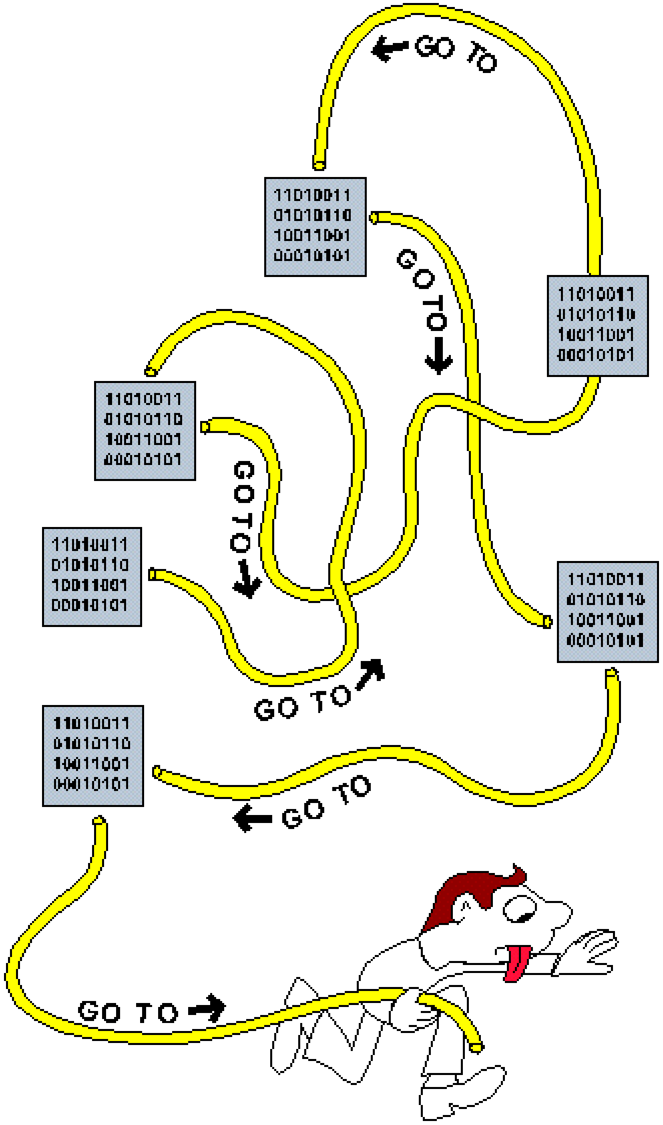
1. Debug data access
2. Maintain data access
3. Optimize data access for performance

“Spaghetti code” – bad thing

https://en.wikipedia.org/wiki/Spaghetti_code

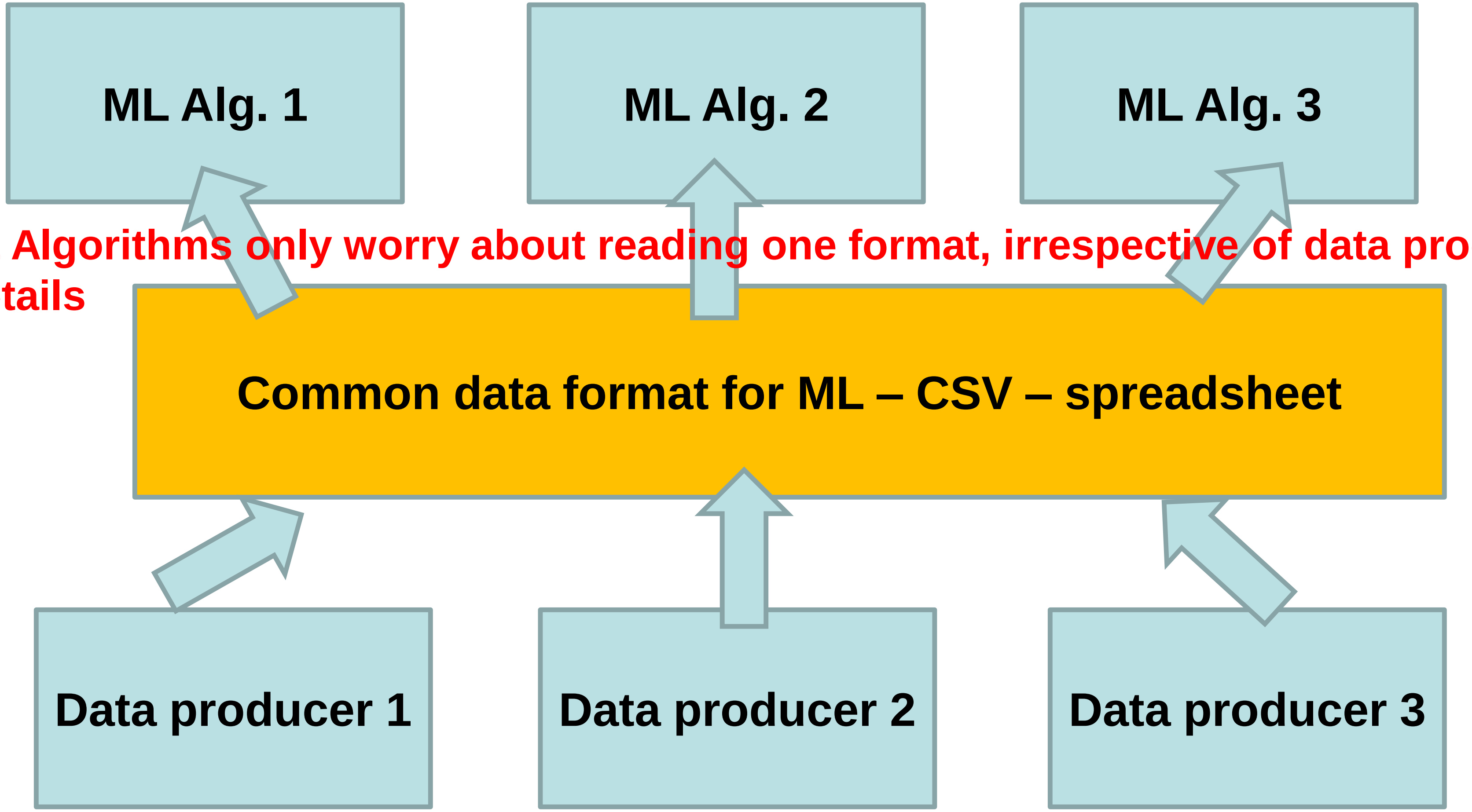
- Code that overuses GOTO statements rather than structured programming constructs, resulting in convoluted and unmaintainable programs, is often called spaghetti code.[2] Such code has a complex and tangled control structure, resulting in a program flow that is conceptually like a bowl of spaghetti, twisted and tangled.[3] In a 1980 publication by the United States National Bureau of Standards, the phrase **spaghetti program** was used to describe older programs having "fragmented and scattered files".[4] Spaghetti code can also describe an anti-pattern in which object-oriented code is written in a procedural style, such as by creating classes whose methods are overly long and messy, or forsaking object oriented concepts like polymorphism.[5] The presence of this form of spaghetti code can significantly reduce the comprehensibility of a system.[6]

- Hard to debug and maintain – big problem



https://www.pcmag.com/encyclopedia/term/spaghetti-code

For data intensive apps like ML/AI achieve modularity by using standard data formats



ML Alg. 1

ML Alg. 2

ML Alg. 3

ML Algorithms only worry about reading one format, irrespective of data producer details

Common data format for ML – CSV – spreadsheet

Data producer 1

Data producer 2

Data producer 3

Data producers only need to know how to generated CSV, they do not worry About Interface to each ML alg.

Architectural design patterns

- Predefined and proven *high level* solutions for particular class of problems
- Usage: *Analyze* your application requirements then try to *map* into one of the known architectural patterns, then evaluate to chose the best
- Examples: Model-View-Controller; Layered Architecture (or Three (Four) Tier Arch.); Repository; Client-Server
- http://en.wikipedia.org/wiki/Architectural_pattern

Architecture design patterns: structure

- Description (not the code)
- Example
- When used
- Advantages
- Disadvantages

Some resources on architectural patterns

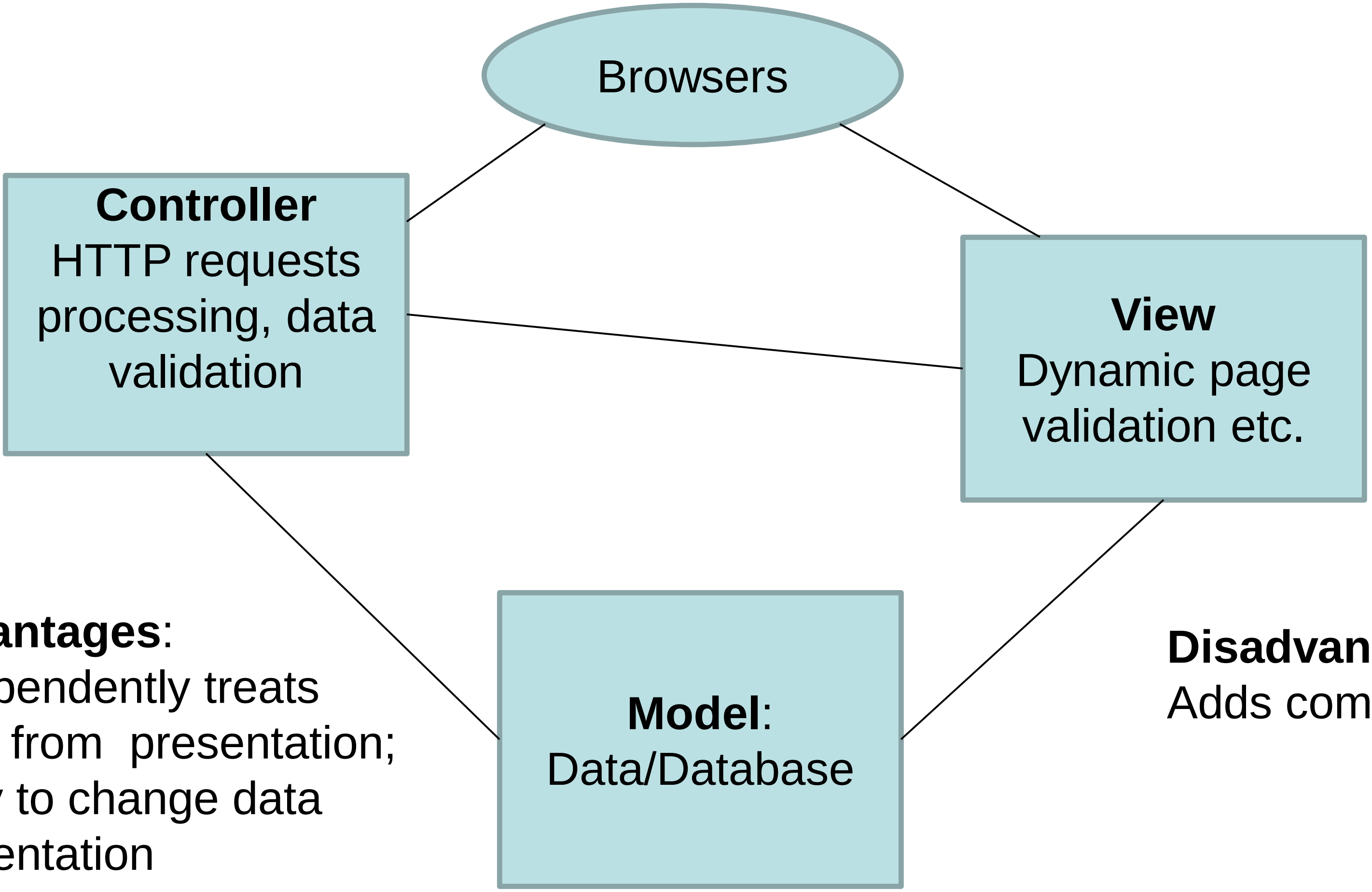
- MVC
  - http://en.wikipedia.org/wiki/MVC_Pattern
- Multitier architectures
  - http://en.wikipedia.org/wiki/3-tier_architecture
- Client server
  - http://en.wikipedia.org/wiki/Client_server
- Blackboard
  - http://en.wikipedia.org/wiki/Blackboard_system

MVC – preferred model/pattern for the team project

- Good explanation with example codes
  - http://www.tomdalling.com/blog/software-design/model-view-controller-explained/

- What each subsystem “needs to know” about others? (The less it knows the more modular it is…)
  - There are some differing definitions as to what model can do and knows about – does it also do some business logic/data interpretation?
  - We prefer that model simply manages data, and business logic is separate (in controller)
    - Why is this good?

Model-View-Controller (in the context of Internet) (adapted from I. Sommerwille)



Browsers

Controller
HTTP requests processing, data validation

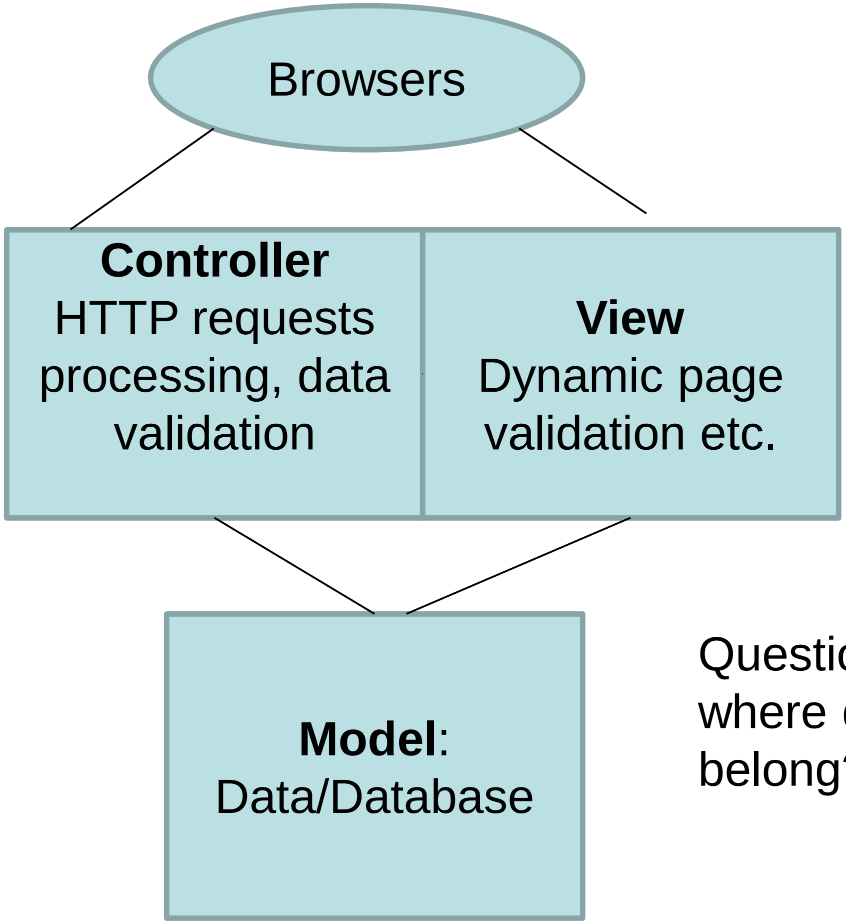
View
Dynamic page validation etc.

Model:
Data/Database

Advantages:
Independently treats Data from presentation;
Easy to change data presentation

Disadvantages:
Adds complexity

Model-View-Controller modification (adapted from I. Sommerwille)



Browsers

Controller
HTTP requests processing, data validation

View
Dynamic page validation etc.

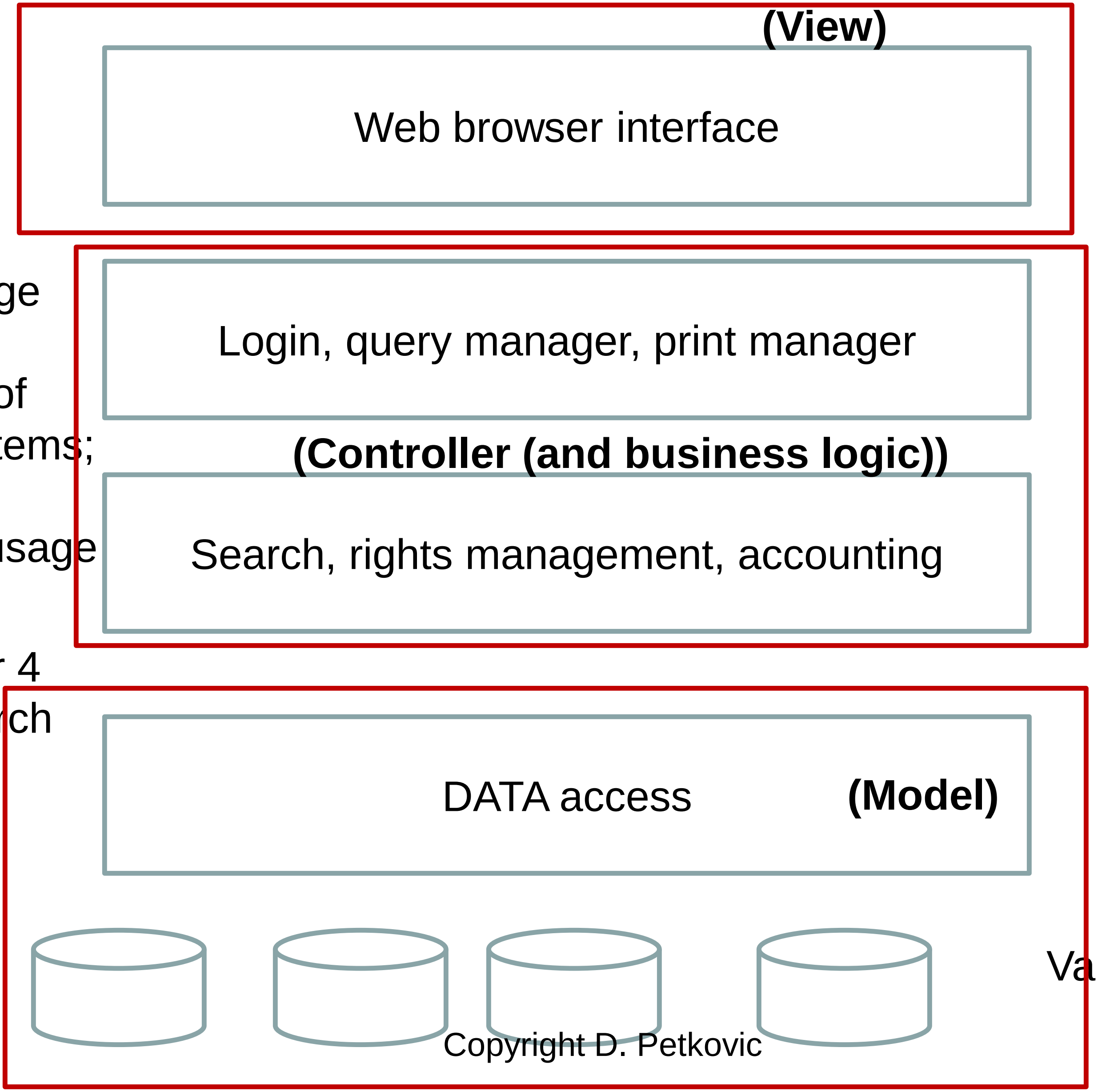
Model:
Data/Database

Question:
where does search code belong?

Layered Architecture: example of Internet based library system (adapted from I. Sommerwille)

Usage:
Complex large systems;
Built on top of Existing systems;
Security;
Formal DB usage

Variant: 3 or 4 tier WWW arch


(View)
Web browser interface
Login, query manager, print manager
(Controller (and business logic))
Search, rights management, accounting
DATA access
(Model)
Various databases

Advantage
Modular;
Easy to interface and replace layers

Disadvantage
Complex;
Not always easy To separate layers

How to communicate/describe high level architecture

- Text (list components)
- Simple block diagrams (informal) – sometimes with special tools
- Unified Modeling Language – UML – formal way

UML – Unified Modeling Language

- **The Object Management Group (OMG) specification states:**
- *"The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components."*

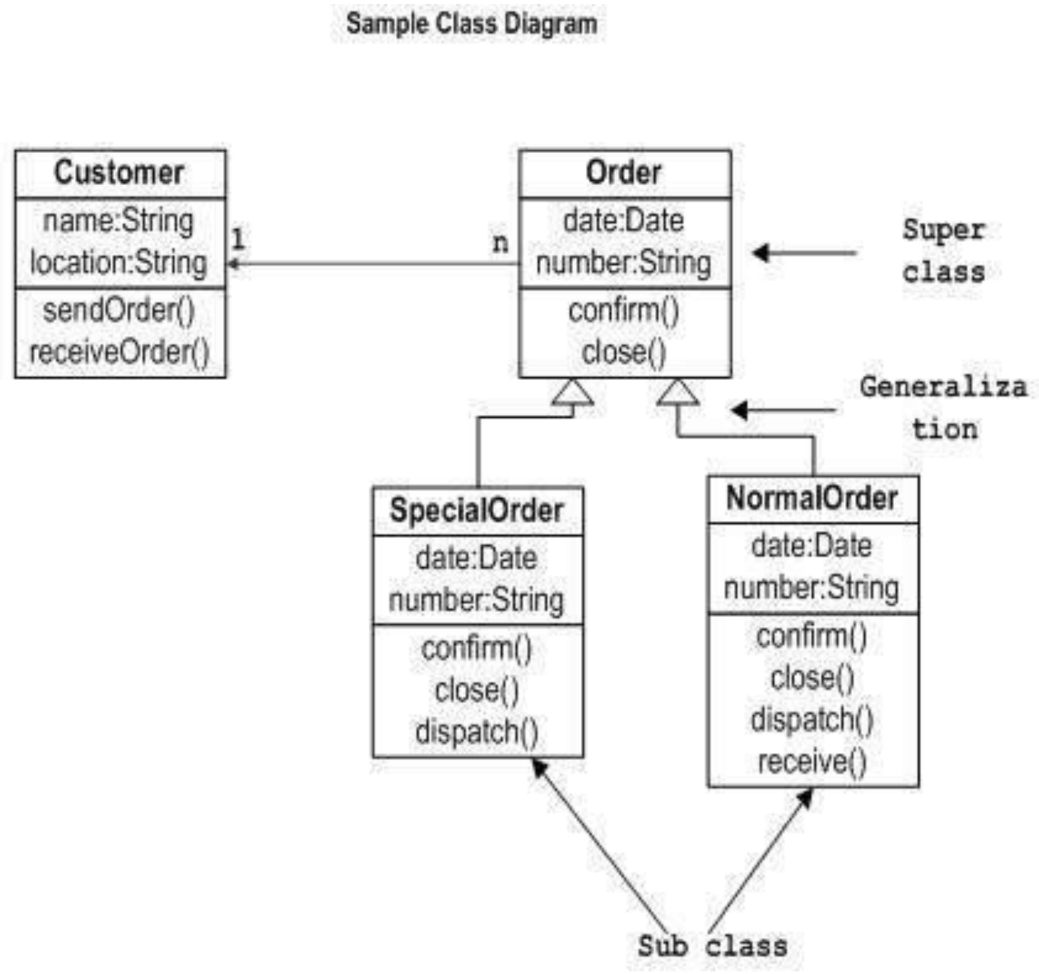
UML tutorials

- https://www.tutorialspoint.com/uml/uml_overview.htm

- Class diagrams
  - https://www.tutorialspoint.com/uml/uml_class_diagram.htm
- Activity diagrams (flowcharts)
  - https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
- State chart diagrams
  - https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm
- Interaction (sequence) diagrams
  - https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm
- Deployment diagrams
  - https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm

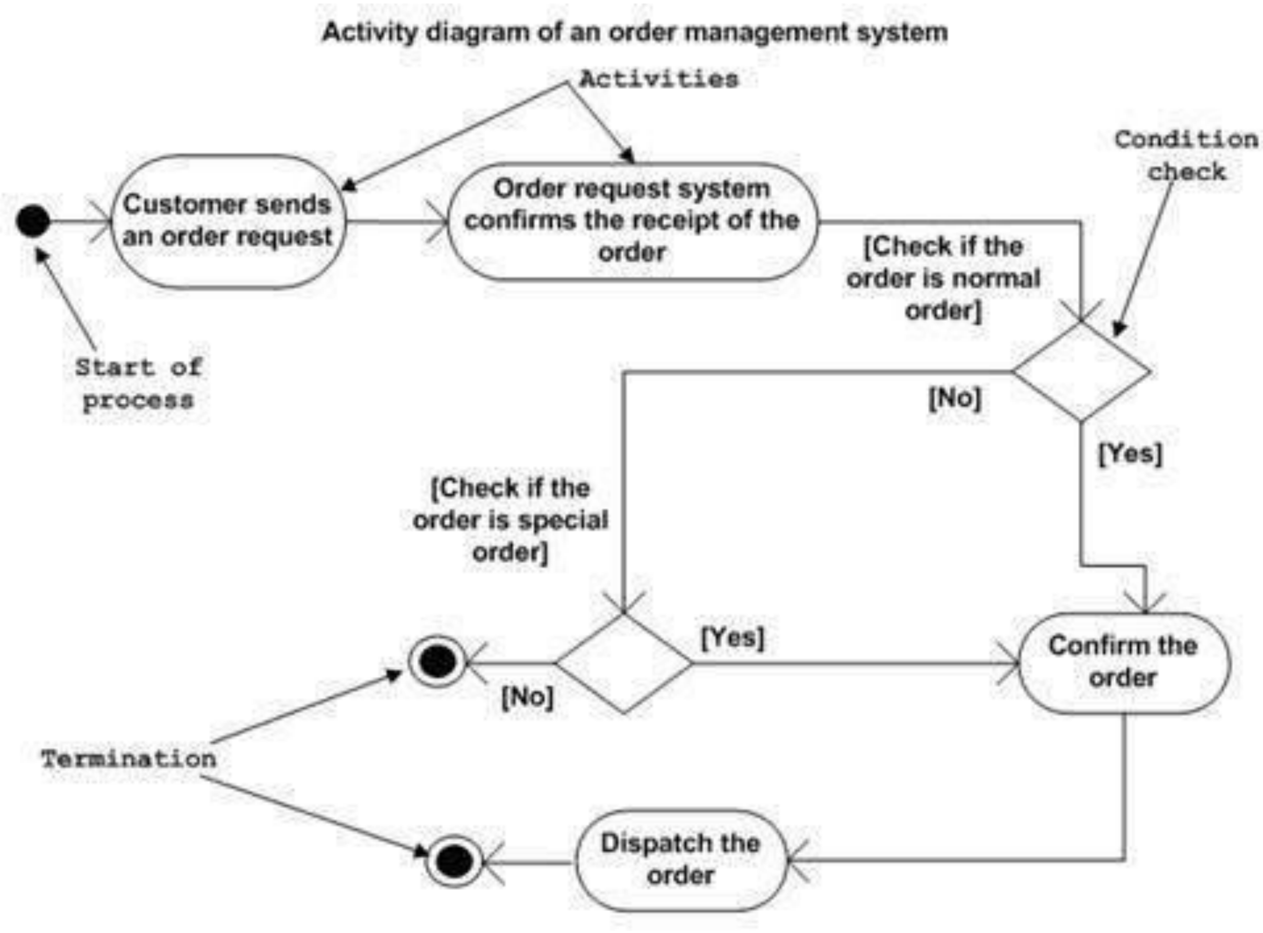
Class diagrams

https://www.tutorialspoint.com/uml/uml_class_diagram.htm



Activity diagrams (flowcharts)

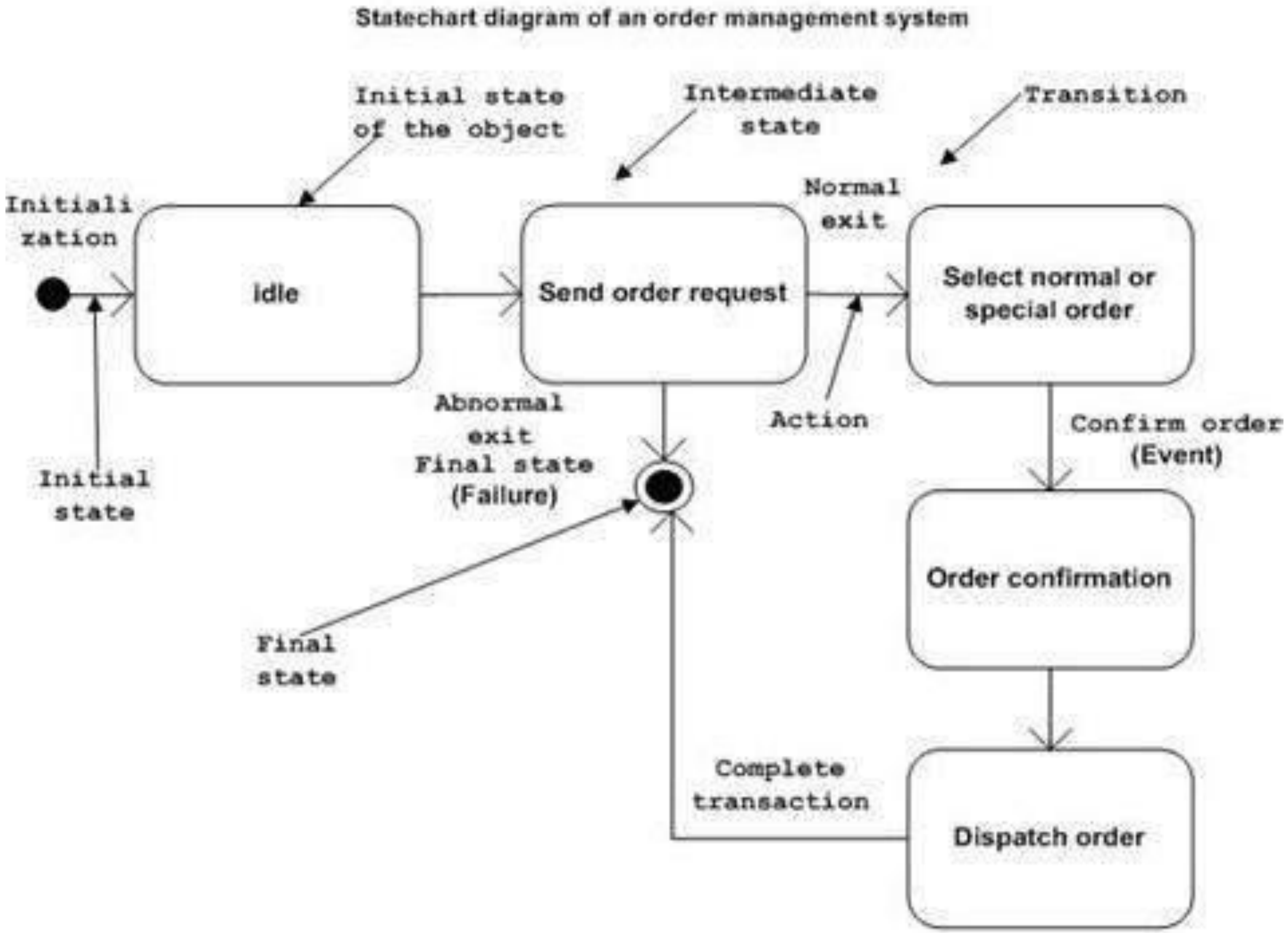
https://www.tutorialspoint.com/uml/uml_activity_diagram.htm


Activity diagram of an order management system
Activities
Condition check
Customer sends an order request
Order request system confirms the receipt of the order
[Check if the order is normal order]
Start of process
[No]
[Yes]
[Check if the order is special order]
[Yes]
[No]
Confirm the order
Termination
Dispatch the order

Copyright D. Petkovic

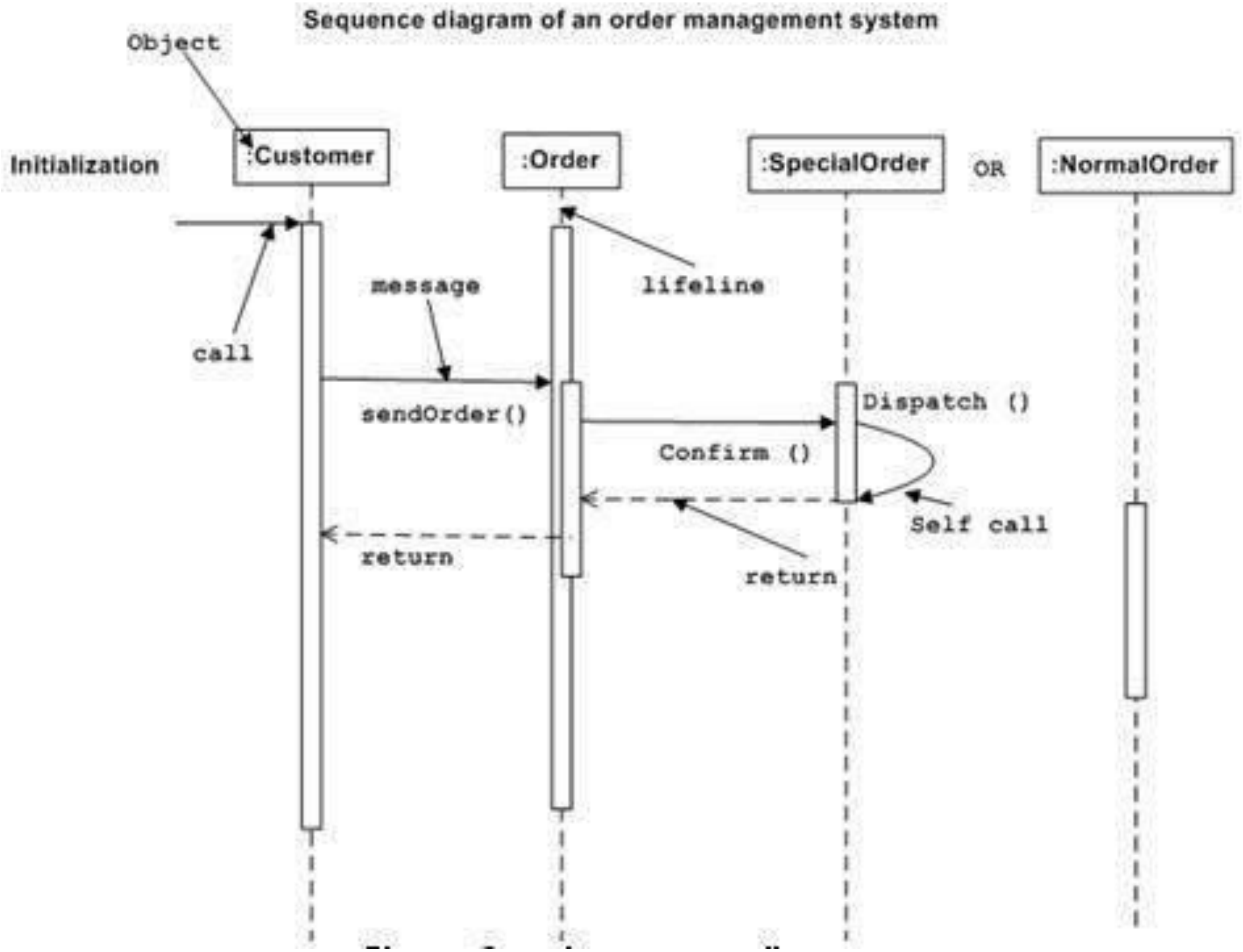
State chart diagrams

https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm



Interaction (sequence) diagrams

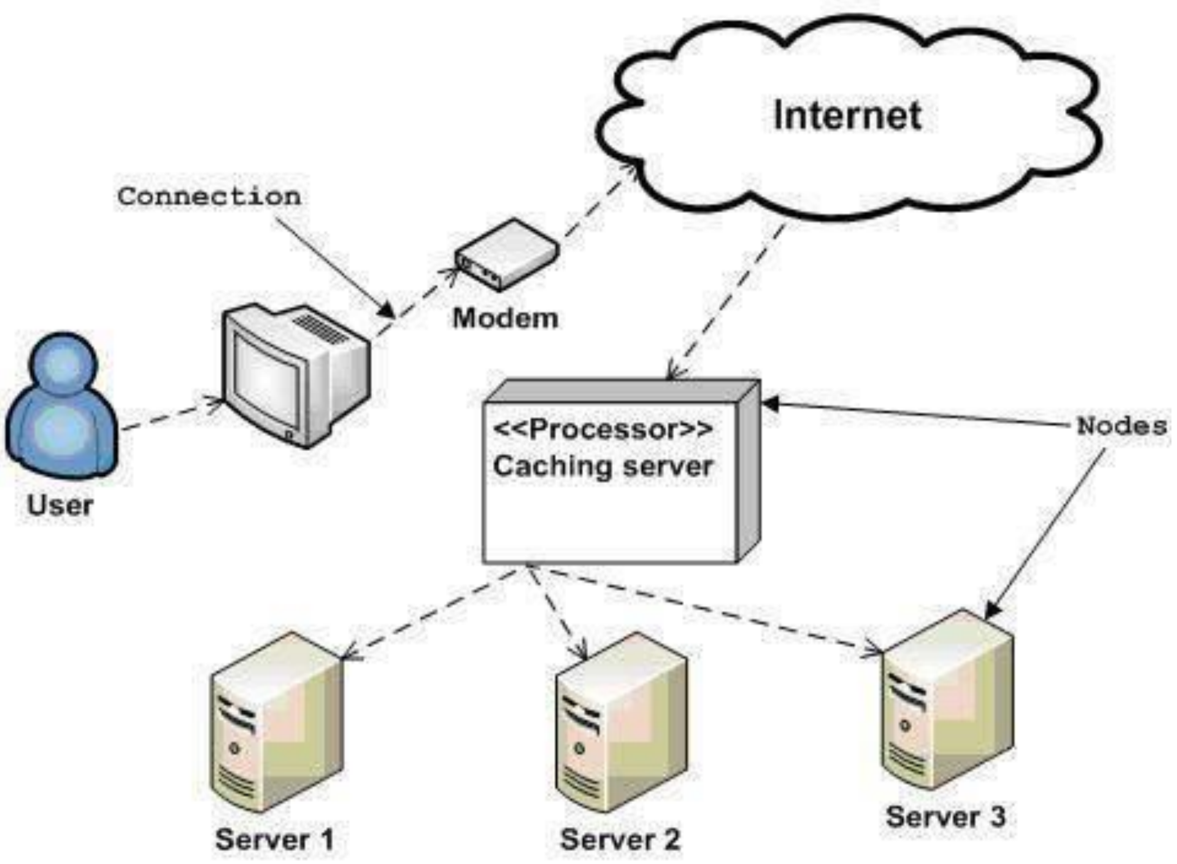
https://www.tutorialspoint.com/uml/uml_interaction_diagram.htm



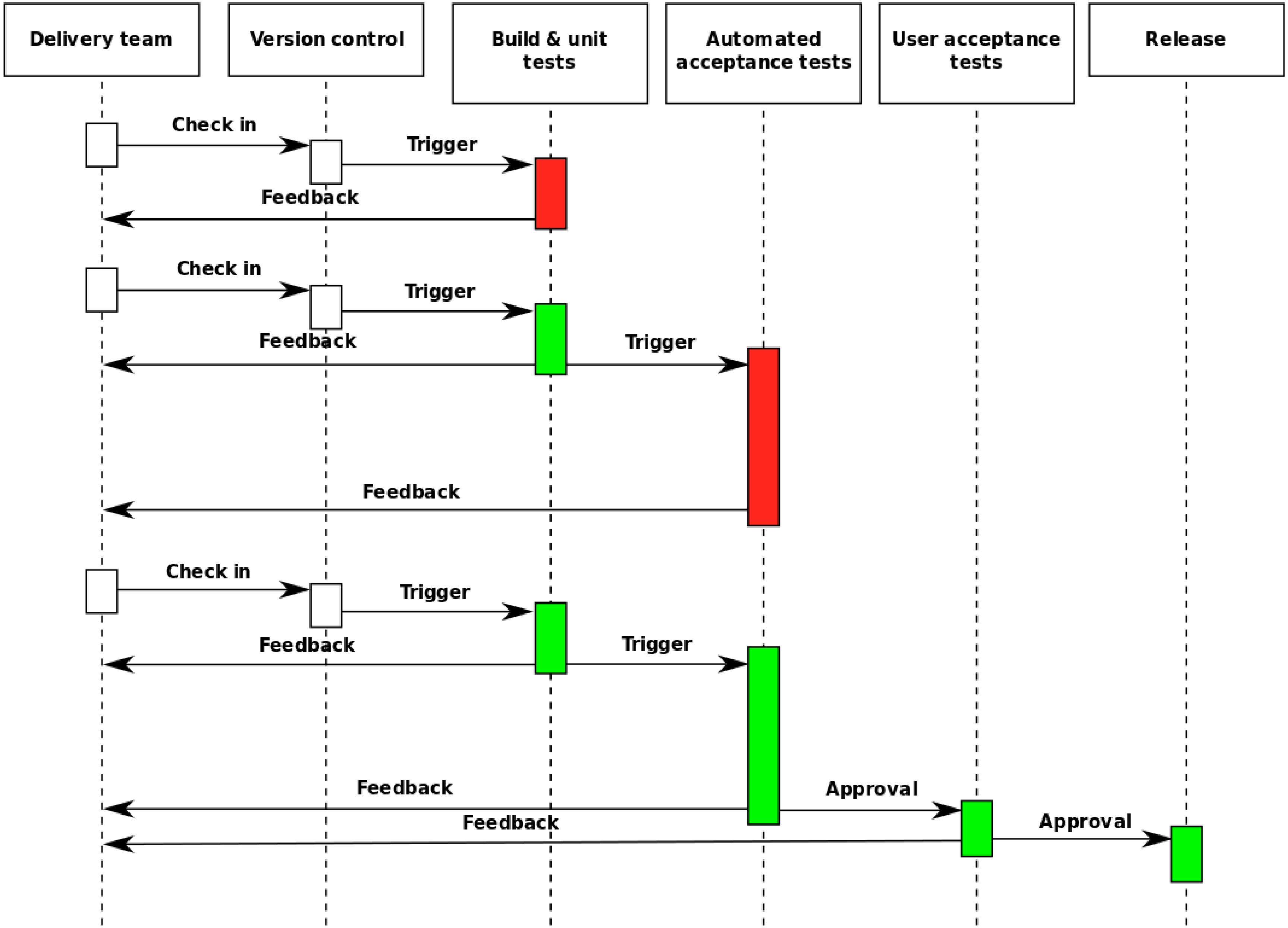
Deployment diagrams

https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm

Deployment diagram of an order management system



UML Seq. diagram used to depict continuous delivery SE process



https://en.wikipedia.org/wiki/Continuous_delivery

SW Frameworks

- Platforms to help develop SW applications – an "environment" including tools, APIs, compilers etc.
- http://en.wikipedia.org/wiki/Software_framework
- Offer tools and environment on top of APIs
- Provide easier ways to build SW systems with the set of prebuilt major functions (tables, login, calendar etc.) and ensures for example cross browser and mobile device compatibility (Bootstrap)
- Commercial and open source solutions available

BUT

Questions to ask re: Frameworks

- Functionality: will it do what I need to do now AND in the future
- Maintainability: how do I maintain the system built by this framework
- Licensing, terms of use, price
- Cross platform issues
- Learning curve of the team to use it
- Availability of support and documentation
- Market share, how widely used, stability of the company producing it, availability of skilled people

Do not believe the marketing – VERIFY: install, try and test it for YOUR application and functions

If you decide to use specific frameworks read documentation and follow instructions (this is recurring problem in our SE class)

Use frameworks properly and do not hack around it!

Ask ChatGPT for inpuit

Importance of SW architectures, QA and good SE processes like code reviews

- Case of COVID-19 modeling SW by Prof. Ferguson in UK
  - https://www.dailymail.co.uk/news/article-8327641/Coronavirus-modelling-Professor-Neil-Ferguson-branded-mess-experts.html
- Critical government policies made based on results from disease spread modeling application
- SW itself was not well developed, implemented nor tested➔ absolutely contrary to best SE practices and what we teach in CSC 648-848
- Failure in:
  - Proper SW architecture design and enforcement
  - No QA
  - No adequate code review

Arch. Design involves a lot of decision making (trait of good SW Eng. As per ACM sldies)

- What to use, which pattern, which technology
- Tradeoffs
- Cost Benefits
- Best ways to do it (there is more than one way)

- Requires experience but also we have ChatGPT assistant to help – use it as advisor but YOU are in charge and responsible

New world of GenAI for SE – BUT architecture still needs to be done by human

- **Architecture of the whole system**: Need human to architect the whole app into modules/subsystems then ask GenAI to help with each subsystem
  - Follow best ***architectural patterns*** like MVC
  - Ensure m***odularity:***
    - ***Intra-subsystem cohesion***: The subsystem has well defined and cohesive function (I.e. data gathering, not data gathering and data processing).
      - Test: can you describe its function in one simple sentence
    - ***High intersystem independence or loose coupling*** (I.e. separate data storage from data rendering)
      - Clean, well defined preferably standard interfaces
    - For data intensive apps use **standard data formats** as interfaces
- Then ask genAI to help with each module

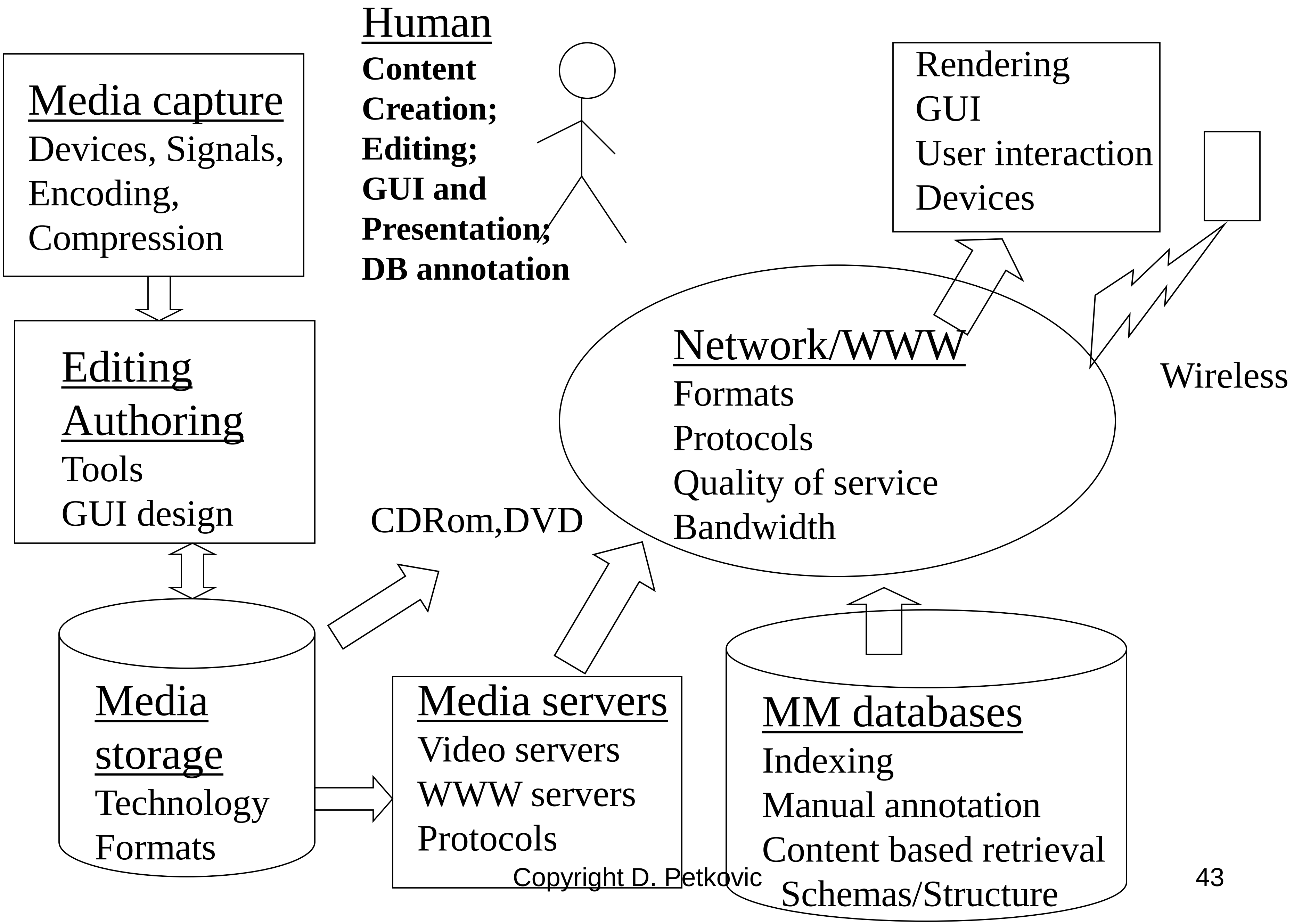
Verify and test!

Multimedia Architectures and Search Design Patterns

Multimedia Information Systems

- Information systems containing, among other types of data, multimedia data
- General uses:
  1. ***Find****: Search and browse* to find the right information
  2. ***View*** the information to make a decision
  3. ***Use*** the info (play, download, burn CD, edit, buy…)
- Requires proper data capture, indexing, storage, delivery and rendering
- Examples: DB of music, videos, training material, e-commerce, marketing, maps

Multimedia "lifecycle"



Media capture

Devices, Signals, Encoding, Compression

Human

Content Creation; Editing; GUI and Presentation; DB annotation

Editing Authoring

Tools

GUI design

Rendering

GUI

User interaction

Devices

Network/WWW

Formats

Protocols

Quality of service

Bandwidth

Wireless

CDRom,DVD

Media storage

Technology

Formats

Media servers

Video servers

WWW servers

Protocols

MM databases

Indexing

Manual annotation

Content based retrieval

Schemas/Structure

MM “programming”

- Content
  - Creation
  - Capture
  - Editing
  - Encoding
- Data organization: Metadata and annotations for DB (data about MM like author, synopsis, title etc.), schemas, DB management
- Presentation, GUI development

Each is very human intensive and involves different skills

Multimedia Data Elements

- *Raw Media*: bits containing content that is played or viewed
  - Color images
  - Graphics
  - Animation
  - Audio
  - Video
- *Supporting data*
  - Used for search (*metadata*)
  - Supporting info (not used for search): Text and supporting information (text, HTML, PDF, PowerPoint, keyframes, thumbnails etc.) to display items (low and high detail)

MM Inf. Systems >> Functionality >> Search

simple

- *Browse by category (*e.g. from the list)
- *Free Text entry field* (possibly with domain selection pulldown)
- *Parametric search e.g.* form-like search (can put value limits on variables like title, price, image size, video duration) – fixed structure
- *Advanced, Boolean* (can create ad-hoc logical combinations of search conditions)
- *Content based retrieval* – search based on automatically extracted information from raw media (give me images "like" this) - R&D topics for now

complex

Search examples

- Alphabetical/list/directory
  - https://www2.eecs.berkeley.edu/Faculty/Lists/CS/faculty.html
- Product – search with categories and filters
  - www.amazon.com
- Media (stock images)
  - http://www.dreamstime.com/?gclid=CKqViLDJyKsCFRAaQgod2Ba21w

Components of search functions

- ***Metadata*** – describe items in terms of text or numerical data which is used for searching. Input to search or indexing algorithm

- ***Index*** - a data structure enabling search, computed off-line during indexing phase. Input: search arguments; Output: pointers to relevant items. Issue: when you add he data do you need to reindex the whole DB or not?

- ***Search arguments*** *(*what is used to search withy) *and* ***scope*** *(*what domain is searched e.g. the whole WWW, local WWW etc*.)*

- ***Search algorithm*** – steps and functions used to do indexing (often called indexing alg.) or search. KW search, text search, SQL, NL etc. Can be done on index (which is pre-computed) or run-time (e.g. running SQL)

- ***Search UI***: how is it offered to the user: a) entering search arguments and scope; b) viewing results; c) refining

Search systems performance measures

- ***Precision****:* our of R retrieved elements, how many (RR) are relevant: RR/R
- ***Recall****:* out of total of NR relevant items in the whole database/domain, how many are retrieved (RR) in specific query: RR/NR
- (Precision is usually easiest to measure in WWW context since NR (total number of relevant items) is usually not known)
- ***Indexing efficiency****:* how long it takes to index the database
- ***Run time efficiency****:* how long does it take to perform search

Example of metadata for movie

- Video format
- Duration
- Title
- Rating
- Time taken
- Author
- Actors
- Director
- Credits
- Brief description
- Number of scenes
- Promotional images
- Movie trailer (type, format, duration)
- Music scores (number,type,format)
- Links to press releases
- Number of scenes

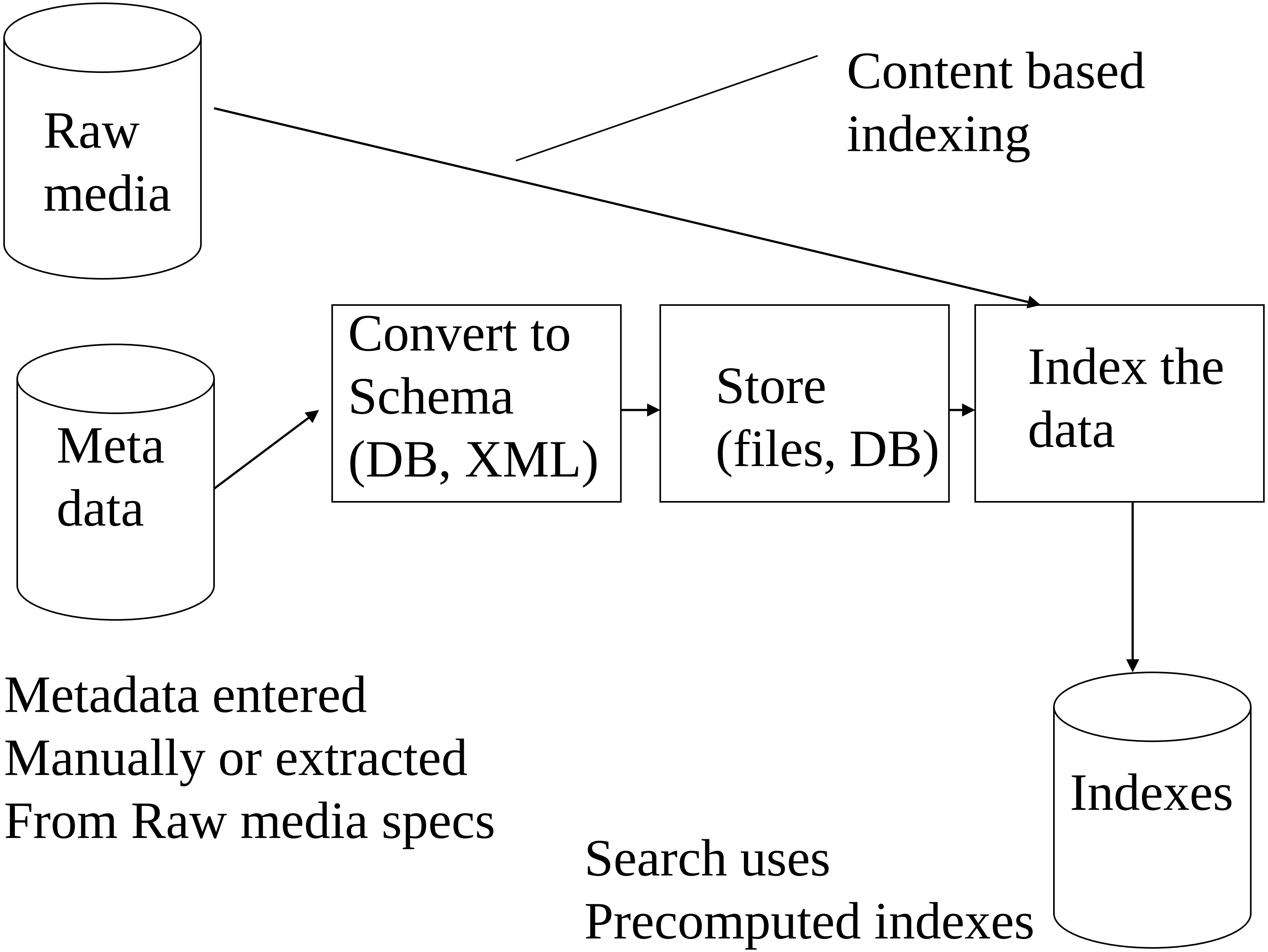
- For each scene
  - Description
  - Actors
  - Duration
  - Keyframes
  - Video

Metadata encoded in XML or DB tables

How to design metadata for media search

- This is more like a librarian and information organization effort and NOT a coding job
- Think of:
  - What users want to search for
  - Characteristics of the media content (movies, images, music) commonly used to search for it
- Avoid too many search fields which are ANDed – most often produces empty result
- Pull-down preferred from free text typing – avoids typos and also refes to existing content

Indexing for MM search



Raw media

Content based indexing

Meta data

Convert to Schema (DB, XML)

Store (files, DB)

Index the data

Indexes

Metadata entered
Manually or extracted
From Raw media specs

Search uses
Precomputed indexes

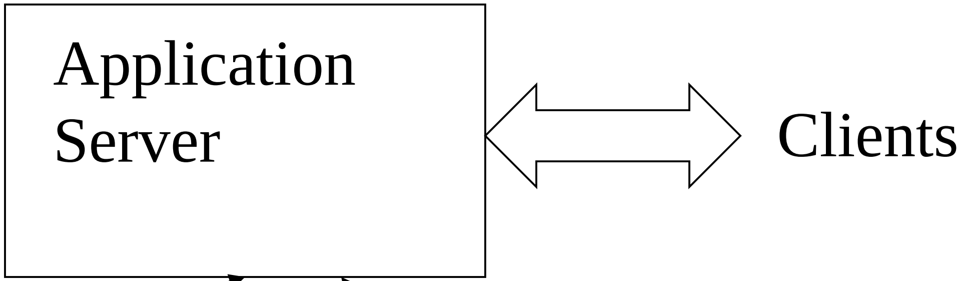
MM Inf. Systems >> Architecture design patterns

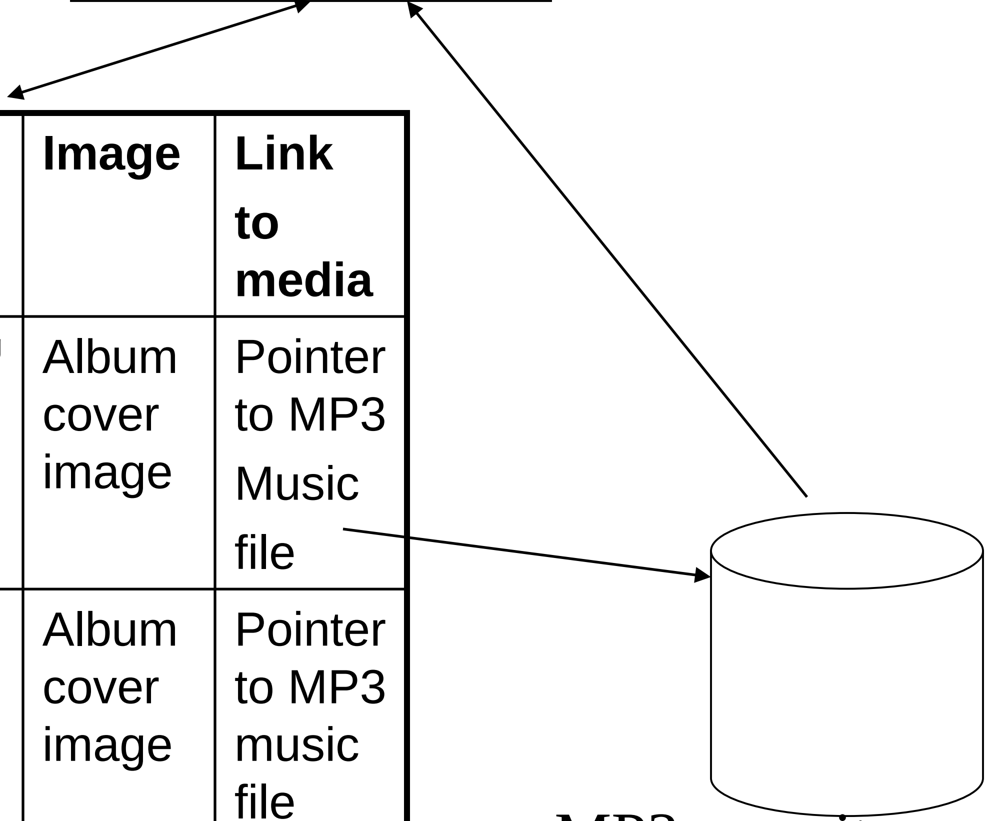
- Quick overview of basic three tier architectures, with emphasis on the WWW
- Specifics related to MM Inf. Systems

MM Inf. Systems Architectures – historical evolution

- Mostly client server or layered arch. with varying degree of data integration in the back-end
  - **Media in standalone on files**, CDRom: collection of interlinked hard wired files; **Traditional DB** with metadata in DB tables and pointers to raw medias on file systems
  - **BLOBs**: Binary Large Objects – integrates media files into the DB
    - https://en.wikipedia.org/wiki/Binary_large_object
    - https://dev.mysql.com/doc/refman/5.7/en/blob.html
  - **Media content management**– full applications buiolt on top of the above
- **Trade-offs**: cost, complexity, administration, performance, security, recovery, manageability etc.

MM Systems With DB and raw media In separate files





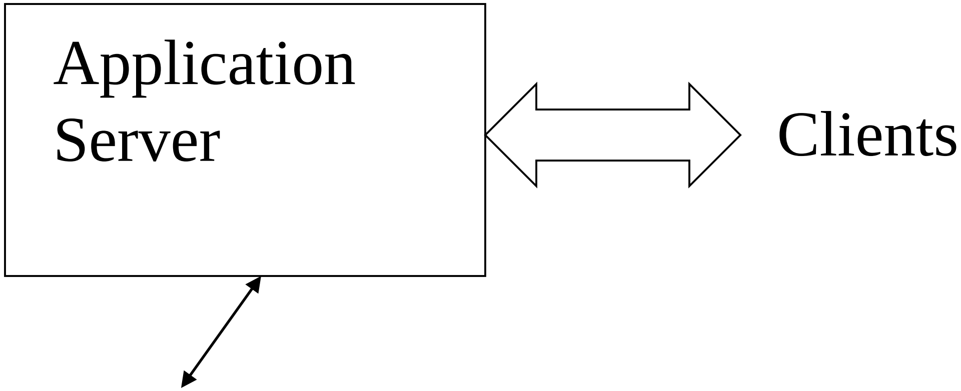
ID	Autho r	Title	Image	Link to media
Item1	Miles Davis	Something Else	Album cover image	Pointer to MP3 Music file
Item2	Rolling stones	Satisfaction	Album cover image	Pointer to MP3 music file

MP3 repository

Table in relational DB

Images can be stored in DB tables or externally

MM Systems
With DB and BLOBS



ID	Author	Title	Image	BLOB
Item1	Miles Davis	Something Else	Album cover image	raw MP3 Music file
Item2	Rolling stones	Satisfaction	Album cover image	Raw MP3 music file

Table in relational DB

MM Inf. Systems with DB and raw media in separate file system

- Pros:
  - Simple
  - Efficient
  - Existing media data kept in place
  - Media data in their natural format
  - Easy to load and access media data –no DB overhead
- Cons:
  - No admin and access control
  - Hard to manage access by many users
  - No transaction security (backup, roll-back etc.)
  - Data on files easily gets out of synch with metadata in DB
  - No leveraging of all the DB tools and functionality such as distribution etc.

BLOBs may Help here

- Files vs. BLOBs is complex and controversial (efficiency, overhead, also deepens on file size)

Key problem in use of classic relational DB for metadata management

- Metadata vary in format and change in time
- Often have messing elements
- Hard to fix it ahead of time (which is required for Relational DB implementation)
- Too brittle and inflexible
- Slow in data ingest/input
- Hard to use in real applications
- BUT….NO-SQL to the rescue

No-SQL for managing metadata

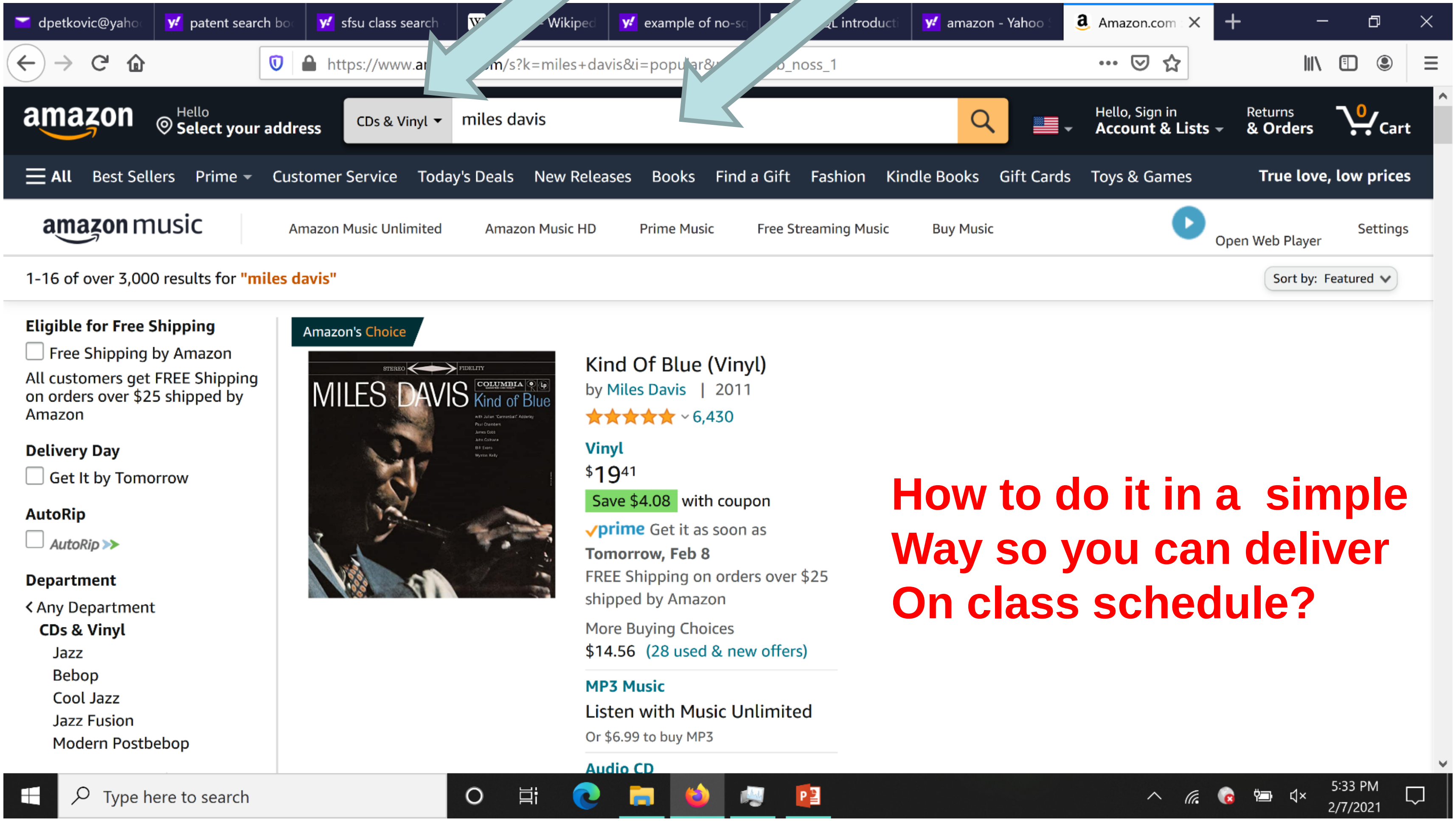
- Driven by Internet world (data harvesting)
- New technology – based on name-value pars -Jason objects
  - https://en.wikipedia.org/wiki/NoSQL
- Coverts data into key-value pairs in strings; fast index used to find data from the key
- Fast ingest (import/input) and search
- Allows SQL access on top of no-SQL
- Not “safe” or ACID as relational DB but this is OK for metadata/internet/search
- Major improvement in for metadata management and for search engine databases – very good high level comparison of SQL and NoSQL for data analytics and search engines
  - https://www.forbes.com/sites/metabrown/2018/03/31/get-to-know-relational-and-nosql-databases-that-power-big-data-analytics/#719d56091943

Some NoSQL resources

- About noSQL
  - https://www.ibm.com/cloud/learn/nosql-databases
  - https://www.w3resource.com/mongodb/nosql.php
- MongoDB
  - https://www.mongodb.com/nosql-explained/examples
- Postgres noSQL
  - https://fulcrum.rocks/blog/why-use-postgresql-database/#:~:text=Postgres%20NoSQL%20also%20enables%20interaction%20with%20other%20sources,Redis%2C%20Neo4j%2C%20Twitter%2C%20LDAP%2C%20File%2C%20Hadoop%20and%20others.

Some specific related to team project

Common search today: Main category (precise hard filter), with narrow down using fuzzy texts search



miles davis

1-16 of over 3,000 results for "miles davis"

Eligible for Free Shipping

□ Free Shipping by Amazon

All customers get FREE Shipping on orders over $25 shipped by Amazon

Delivery Day

□ Get It by Tomorrow

AutoRip

□ AutoRip

Department

‹ Any Department

CDs & Vinyl

Jazz

Bebop

Cool Jazz

Jazz Fusion

Modern Postbebop

Kind Of Blue (Vinyl)

by Miles Davis | 2011

★★★★★ 6,430

Vinyl

$19.41

Save $4.08 with coupon

prime Get it as soon as Tomorrow, Feb 8

FREE Shipping on orders over $25 shipped by Amazon

More Buying Choices

$14.56 (28 used & new offers)

MP3 Music

Listen with Music Unlimited

Or $6.99 to buy MP3

Audio CD

How to do it in a simple Way so you can deliver On class schedule?

Backend for search combining categories (exact) and text (substring) search (like Amazon search)

- Case study: you want to search items by combining:
  - Category (e.g. furniture, electronics…) WITH
  - Fuzzy text search on text (e.g. concatenate item title, description in one DB field (e.g "SOFA red sofa by Lui XVI")
- One simple solution (OK for small data size):
  - SQL precise search for *categories* from DB column called *categories ANDed with*
  - *%like* search on *text field* e.g. item description + item title
    - http://www.mysqltutorial.org/mysql-like/
  - MySQL and other DB have more text search options
    - https://dev.mysql.com/doc/refman/5.7/en/fulltext-search.html
- ***(In many applications like product or service searching fuzzy and approximate search are desired even if it creates some false positives – think of a good salesperson – always shows more to chose from)***

Suggested DB organization when media item (image, video, sales item, real-estate post etc.) has associated *category used for search*

- Example: amazon.com – main Category (E.G. SHOES) combined with free texts earch
- Have a basic DB table with media ID, title, decryption etc. and CATEGORY as a foreign key
  - Media data referenced as pointer to file or BLOB
- Have separate DB table for CATEGORY
- Benefits:
  - If categories are changed or edited you change only DB CATEGORY table
  - DB CATEGORY table drives all UI menus (search, upload) – much easier to manage vs. hard coding categories in java script

Suggestions and design patterns for search – important for team project

- For UX: Use market leading apps as a guidance – leverage what people are used to! Can consult ChatGPT too
  - Most popular today: main category as pull down menu (top level filter) attached to fuzzy text entry search field - ANDed together (e.g. Amazon)
  - In search results always say how many total found – upper left
  - Have search always there, as part of CSS and nav bar – user should be able to search any time
  - Make search input persistent – stays always there until user changes ➔ keeps the context
- Validate search text input – limit it to some small size say 100 characters to prevent code injection
- "Own" error message – **you design them with user in mind!**
  - If text entry not valid say so and advise user how to correct
  - If no items found never waste the screen – tell users to revise the query and show some related items

How to store image paths in DB column

- Avoid absolute hard coded path names – makes your app very “brittle” and sensitive to changes in deployment directory
- Use relative paths e.g. store image name and append relative path to the root of appl. or current work folder
- Some discussion is here
  - https://www.daniweb.com/programming/databases/threads/475006/storing-image-file-paths-into-database
  - https://www.quora.com/What-is-the-best-way-to-store-100-images-in-a-MySQL-database-in-this-case
- Can also use UUID – universally unique identifier
  - https://en.wikipedia.org/wiki/Universally_unique_identifier

More on efficient image handling – efficient browsing is important! (By Anthony Souza)

- Ensure that you have both full resolution image for result details (but limit it to a few Mbytes) as well as *thumbnail image* (smaller – about 20 Kbytes) for results list display
- In your item database have pointers (relative paths) for both full res and thumbnail image
- Create thumbnail image automatically upon user posting of item and image data, use Sharp module in Node, in Python, you can use the Python Pillow package(PIL)
- Image (raw data files) storing and serving to the browser for view or search results
  - Simply save the images in the file system where your app is running (make sure file system is secured)
  - In the DB (e.g. for each product item) you would store the file paths for these images.
  - Send links to these images and not the images themselves and then have either a web server or node/python serve these images. In node/express this can be done with the express.static middleware. In python/flask you can use the built-in static endpoint or a better solution for flask at least is to put it behind a web server and let the web server itself serve the static files.

Our own tutorials for app infrastructure/architecture (documented code) – feel free to use and modify

- Tutorial with nodejs, developed by our former student and TA Nicholas Stepanov
  - https://medium.com/@nicholasstepanov/search-your-server-side-mysql-database-from-node-js-website-400cd68049fa
- Tutorial with flask, developed by our SE instructor Jose Ortiz
  - https://medium.com/@joseortizcosta/search-utility-with-flask-and-mysql-60bb8ee83dad
- Tutorial with PHP, by Jose Ortiz
  - https://medium.com/@joseortizcosta/search-utility-with-php-and-mysql-as-backend-server-technologies-d3dac5128d8
- How to use:
  - Study code to learn
  - Customize for your app; deeply and test then put on master branch
  - Document well, establish good APIs
  - Use as templates/architecture to guide each team member
  - Perform constant code reviews to ensure people follow the templates and APIs

How to provision the DB at high level – from our CTO Anthony

- It is NOT a requirement to have your database and application on different servers. as in industry ➔ make it simple

Main options

- Main production DB and local DB for each member
  - Each member deploys and maintains their own localhost database (e.g. on their laptops) and the team has one main production DB for your application.
  - However, keeping these databases consistent is going to be a challenging task
- Main DB and Test DB on deployment server, team accesses them
  - Run 2 databases (under the same DBMS software) on the remote server: production DB and a Test db.
  - Both should be maintained by the same team member.
  - The test and production database should be a similar as possible.

GenAI tools

- Check class slides…..
- Use it but review code, and test