Architectural Style

1. What is an architectural style
   1. An architectural style the highest level of abstraction of a software system. It provides a structure for a software system by organizing it
   2. Into abstract modules. Architectural styles enforce rules and behaviours for the system as to better partition the system and solve recurring problems
2. Why are they important?
   1. Expected behaviour
   2. Time and Cost
   3. Project Success
3. There are many architectural styles
   1. Blackboard
   2. Client – Server
   3. Shared nothing
   4. Event-Driven
   5. N –Tier
4. Why should we examine specific architectural styles?
   1. The point of examining the five chosen architectural styles is not just explain them, but rather to give insight into how to choose the proper architectural style for a new project by analyzing its advantages and disadvantages. Choosing the right style for a project an help avoid failure
5. Research Flow
   1. Context
      1. The problem the architectural style was made to solve
   2. Setup/implementation
      1. How the architectural style is setup and implemented
   3. Advantages/disadvantages
      1. What the advantages and disadvantages are to use the architectural style
   4. Applications
      1. What real world systems use the architectural style
6. Client – Server
   1. Context
      1. Host (Server) receives request from users (client), and sends appropriate response
   2. Implementation
      1. Replaced file server with a database server
         1. Instead of sending back files, it would return queries
         2. Handled by a RDBM system
      2. Uses GUI instead of terminal
      3. Reduced network traffic
      4. Increased scalability
      5. Different versions like 2-Tier, N-Tier, etc.
   3. Advantages/Dis
      1. Advantages
         1. Centralization of control
         2. Improved
            1. Usability
            2. Flexibility
            3. Interoperability
            4. Scalability
      2. Disadvantages
         1. Not robust: if server goes down, client/..
         2. Can be costly
   4. Application
      1. Email
      2. Network printing
      3. WWW
      4. Banking online
7. N-Tier
   1. Context
      1. The 2-Tier client-server architecture presented a problem: when the logic of the application was changed, the client’s application had to be changed as well as tested and redistributed
      2. The 3-Tier architecture was created to separate the different aspects and solve the problem presented by 2-Tier Client-Server
   2. Implementation
      1. Separate the software processing, presentation and data management
      2. Presentation tier provides and interface for the user to communicate with the application
      3. Logic tier manages the logic of the application as well as communicate with the presentation and data tiers
      4. Data tier stores and retrieves information in database or file system
   3. Advantages and dis
      1. Advantages
         1. Work done in one section will not affect other sections
         2. Resources are not shared and so services perform efficiently
         3. Application is scalable because more resources can be added without affecting the others
      2. Disadvantages
         1. Using a large number of tiers adds
            1. More complexity to the software
            2. More cost to hardware and maintenance
         2. Fast network bandwidth and hardware are needed as multiple tiers will be work with
   4. Applications
      1. Website development
8. Blackboard
   1. Context
      1. Made in 40 years ago to solve non-deterministic problems
      2. It replicates the life model of a group of people solving a problem on a blackboard
      3. Each person adds their expertise about that problem on the blackboard and together they discover a solution
   2. Implementation
      1. Knowledge sources
         1. Solve a problem by writing their specific expertise about the problem on the blackboard
      2. Blackboard
         1. A shared database among the knowledge sources that contains problems, facts, and solutions
      3. Control shell
         1. Assigns the proper knowledge source a problem from the blackboard
   3. Advantages and Dis
      1. Advantages
         1. All information in a central location
         2. Easily extendable
         3. Knowledge system are kept independent of each other
      2. Disadvantages
         1. The control shell is hard to implement
            1. Hard to know which knowledge source should tackle the problem
            2. Adding more knowledge sources, adds more complexity to control shell
   4. Applications
      1. Self-driving wheelchair using the Blackboard Architectural style
      2. Sensor handler
      3. Corridor recognizer
      4. Collision detector
      5. Drive controller
9. Shared Nothing
   1. Context
      1. It solves the problem of scalability, a problem often faced by web site
      2. Shared-Nothing splits data across a set of machines such that each machine has complete autonomy over its own task
   2. Implementation
      1. Shared nothing is based on a collection of independent nodes that have their own hard dish drive, memory and I/O interfaces
      2. Each node has total autonomy over its own data and doesn’t share resources with other nodes
      3. To ensure up to data, information is made available to at least two other nodes
   3. Advantages/dis
      1. Advantages
         1. Increases scalability b/c there is no resource contention like in shared-memory architecture
         2. Eliminates a single point of failure
         3. Distributed locking is not an issue since each machine has its own task to perform
      2. Dis
         1. In the context of Databases, a Shared-nothing architecture can cause poor performance of inner-joins over a large pool of data sets
   4. Applications
      1. Web application
10. Event – Driven
    1. Context
       1. It was made to facilitate to systems that need to sense and respond to real time events
       2. Commonly used for interactive GUI systems and for automatic maintenance of systems
    2. Implementation
       1. Event-Driven Architecture can be implemented in 2 ways
          1. Publish/subscribe
          2. Event streaming
       2. 3 components
          1. Event generator
             1. The action that triggers an event
          2. Event channel
             1. Carries the generated event to the event processing engine
          3. Event processing engine
             1. Processes events in one of 3 ways

Simple event processing

Event stream processing

Complex event processing

* 1. Advantages and dis
     1. Advantage
        1. Works well for real-time interactive system
        2. More than one system can be notified of single event
        3. Scalable
     2. Disadvantages
        1. The one major disadvantage to this architecture is that it can cause the system to be slow, as it needs to constantly be checking if an event has occurred
     3. Applications
        1. Game software
        2. Automated maintenance