

Software architecture

▼ Introduction

▼ What



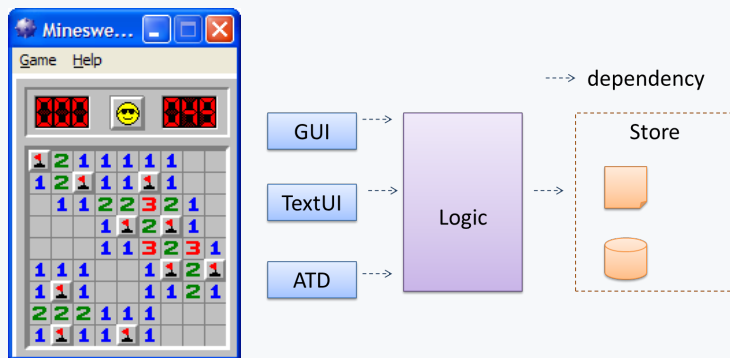
Can explain Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. Architecture is concerned with the public side of interfaces; private details of elements—details having to do solely with internal implementation—are not architectural. ...

Software Architecture in Practice (2nd edition), Bass, Clements, and Kazman

The software architecture shows the overall organization of the system and can be viewed as a very high-level design. It usually consists of a set of interacting components that fit together to achieve the required functionality. It should be a simple and technically viable structure that is well-understood and agreed-upon by everyone in the development team, and it forms the basis for the implementation.

📦 A possible architecture for a *Minesweeper* game:



Main components:

- **GUI** : Graphical user interface
- **TextUi** : Textual user interface
- **ATD** : An automated test driver used for testing the game logic
- **Logic** : Computation and logic of the game
- **Store** : Storage and retrieval of game data (high scores etc.)

The architecture is typically designed by the *software architect*, who provides the technical vision of the system and makes high-level (i.e. architecture-level) technical decisions about the project.

📖 Exercises

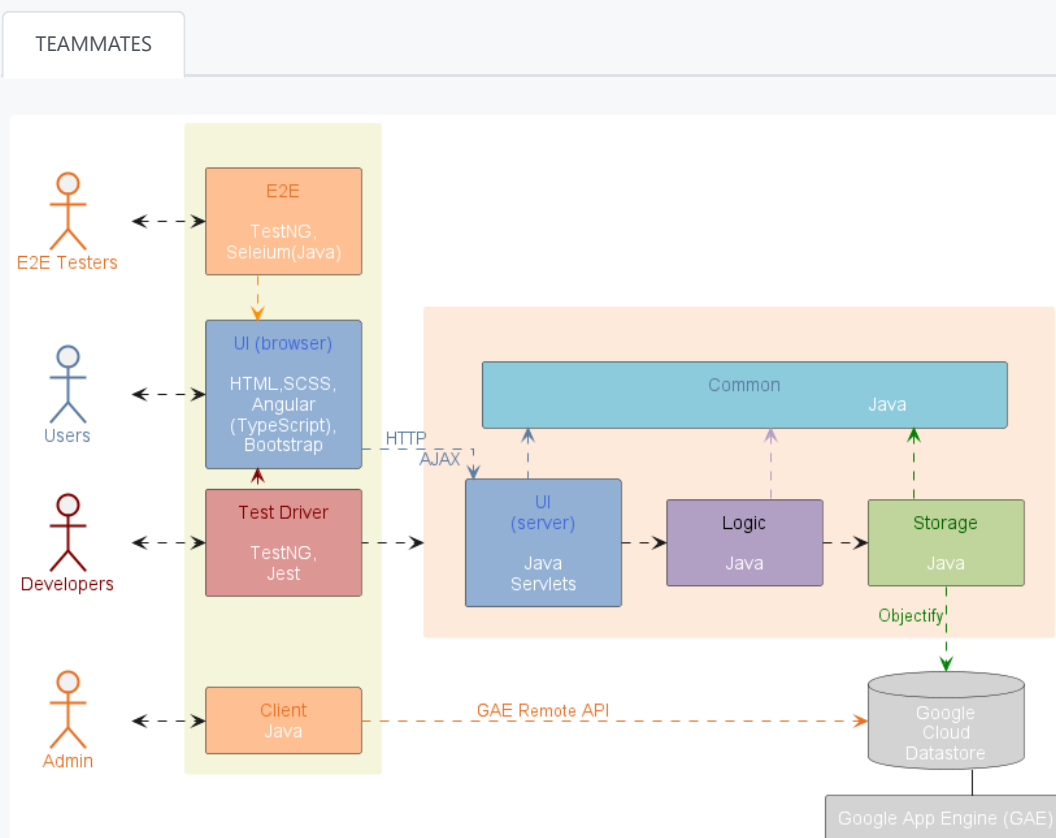


▼ Architecture diagrams

▼ Reading

Architecture diagrams are free-form diagrams. There is no universally adopted standard notation for architecture diagrams. Any symbols that reasonably describe the architecture may be used.

Some example architecture diagrams:



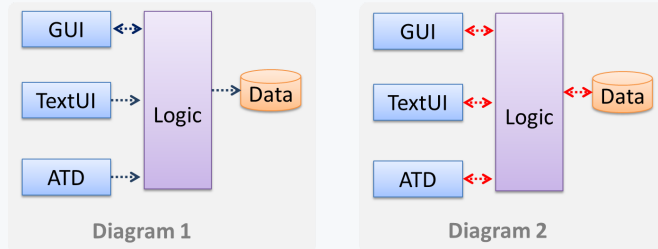
▼ Drawing

While architecture diagrams have no standard notation, try to follow these basic guidelines when drawing them.

- Minimize the variety of symbols. If the symbols you choose do not have widely-understood meanings e.g. A drum symbol is widely-understood as representing a database, explain their meaning.
- Avoid the indiscriminate use of double-headed arrows to show interactions between components.

Consider the two architecture diagrams of the same software given below. Because **Diagram 2** uses double-headed arrows, the important fact that GUI has a bidirectional dependency with the Logic component is no longer captured.

.....> dependency



▼ Architectural styles

▼ Introduction

▼ What

★★★★☆ 🏆 Can explain architectural styles

Software architectures follow various high-level styles (aka *architectural patterns*), just like how building architectures follow various architecture styles.

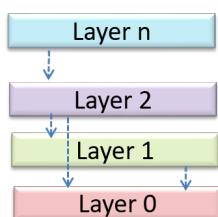
📦 n-tier style, client-server style, event-driven style, transaction processing style, service-oriented style, pipes-and-filters style, message-driven style, broker style, ...

▼ N-tier architectural style

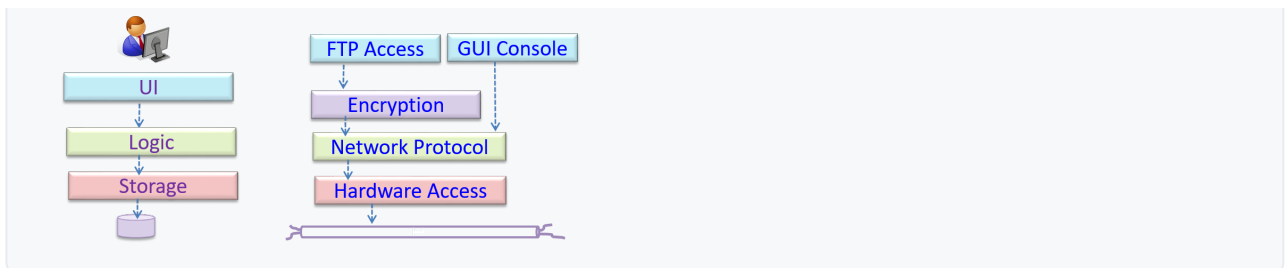
▼ What

★★★☆☆ 🏆 Can identify n-tier architectural style

In the *n-tier* style, higher layers make use of services provided by lower layers. Lower layers are independent of higher layers. Other names: *multi-layered*, *layered*.



📦 Operating systems and network communication software often use n-tier style.

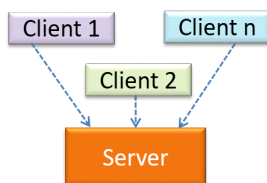


▼ Client-server architectural style

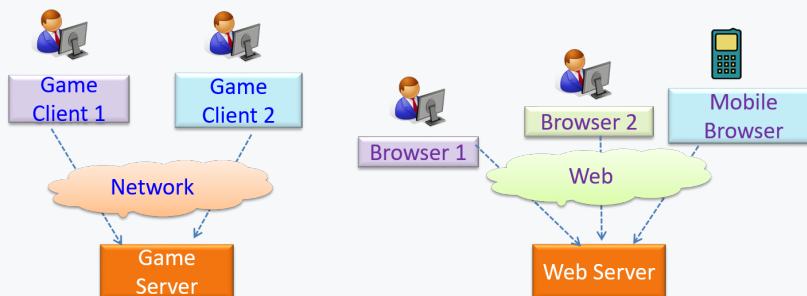
▼ What

★★★★☆ 🏆 Can identify the client-server architectural style

The **client-server** style has at least one component playing the role of a server and at least one client component accessing the services of the server. This is an architectural style used often in distributed applications.



📦 The online game and the web application below use the client-server style.

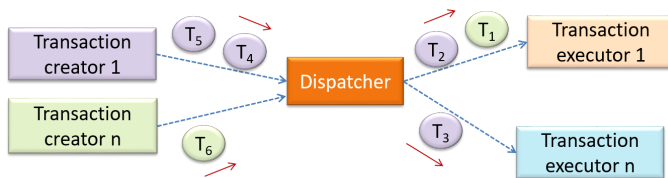


▼ Transaction processing architectural style

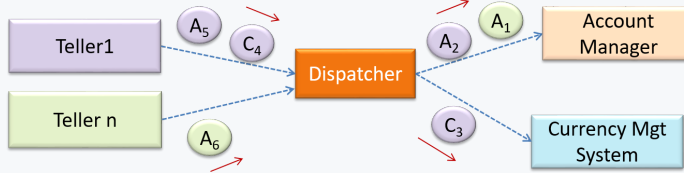
▼ What

★★★★☆ 🏆 Can identify transaction processing architectural style

The **transaction processing** style divides the workload of the system down to a number of **transactions** which are then given to a **dispatcher** that controls the execution of each transaction. Task queuing, ordering, undo etc. are handled by the dispatcher.



📦 In this example from a banking system, transactions are generated by the terminals used by tellers, which are then sent to a central dispatching unit, which in turn dispatches the transactions to various other units to execute.



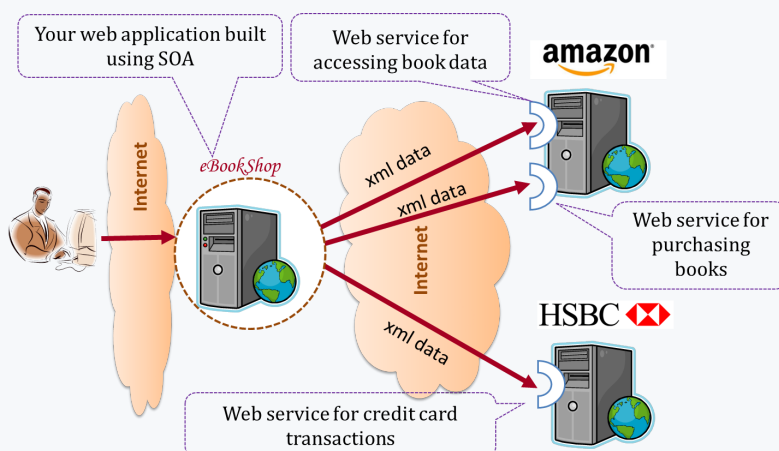
▼ Service-oriented architectural style

▼ What

★★★★☆ 🏆 Can identify service-oriented architectural style

The **service-oriented architecture (SOA)** style builds applications by combining functionalities packaged as **programmatically accessible services**. SOA aims to achieve interoperability between distributed services, which may not even be implemented using the same programming language. A common way to implement SOA is through the use of *XML web services* where the web is used as the medium for the services to interact, and XML is used as the language of communication between service providers and service users.

📦 Suppose that Amazon.com provides a web service for customers to browse and buy merchandise, while HSBC provides a web service for merchants to charge HSBC credit cards. Using these web services, an 'eBookShop' web application can be developed that allows HSBC customers to buy merchandise from Amazon and pay for them using HSBC credit cards. Because both Amazon and HSBC services follow the SOA architecture, their web services can be reused by the web application, even if all three systems use different programming platforms.

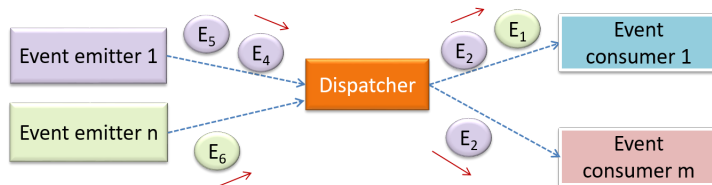


▼ Event-driven architectural style

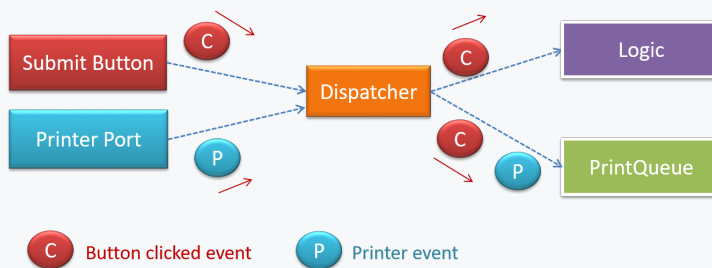
▼ What

☆☆☆☆ 🏆 Can identify event-driven architectural style

Event-driven style controls the flow of the application by detecting events from event *emitters* and communicating those events to interested event *consumers*. This architectural style is often used in GUIs.



📦 When the 'button clicked' event occurs in a GUI, that event can be transmitted to components that are interested in reacting to that event. Similarly, events detected at a printer port can be transmitted to components related to operating the printer. The same event can be sent to multiple consumers too.



▼ More

▼ More Styles

☆☆☆☆ 🏆 Can name several other architecture styles

Other well-known architectural styles include the *pipes-and-filters architecture*, the *broker architecture*, the *peer-to-peer architecture*, and the *message-oriented architecture*.

🔗 Resources

- [Pipes and Filters pattern](#) -- an article from Microsoft about the pipes and filters architectural style
- [Broker pattern](#) -- Wikipedia article on the broker architectural style
- [Peer-to-peer pattern](#) -- Wikipedia article on the P2P architectural style
- [Message-driven processing](#) -- a post by Margaret Rouse



▼ Using Styles

★★★★☆ 🏆 Can explain how architectural styles are combined

Most applications use a mix of these architectural styles.

📦 An application can use a client-server architecture where the server component comprises several layers, i.e. it uses the n-tier architecture.

📖 Exercises

