**1.JAVA is platform independent:**

The byte code generated by the javac compiler can be executed on all kinds of Operating Systems which is unlike other programming languages. Basically, the byte code generated by Java needs JVM only. It does not depend on the operating system. The javac compiler compiles it and makes a .class file or the byte code.JVM OS dependent but .class file or bytecode only on jvm.

**2.Applications**

**Standalone application**-one time install application on every machine.eg media player,anti virus etc

Awt and swing in java used .

1. Features of java.
2. Automatic garbage collection.
3. Portable.
4. Java is a write once, run anywhere.

**Jvm** -provides runtime environment .Does not present physically. Class file to machine code.

**Jre**- contains jvm and other files + libraries. It also provides run time environment. Present physically.

**Jdk -**contains jre and jvm. All java file interpretation is done in jdk.

**3.Java native interface**- Java Native Interface (JNI) is a standard programming interface for writing Java native methods and embedding the Java virtual machine into native applications. The primary goal is binary compatibility of native method libraries across all Java virtual machine implementations on a given platform.

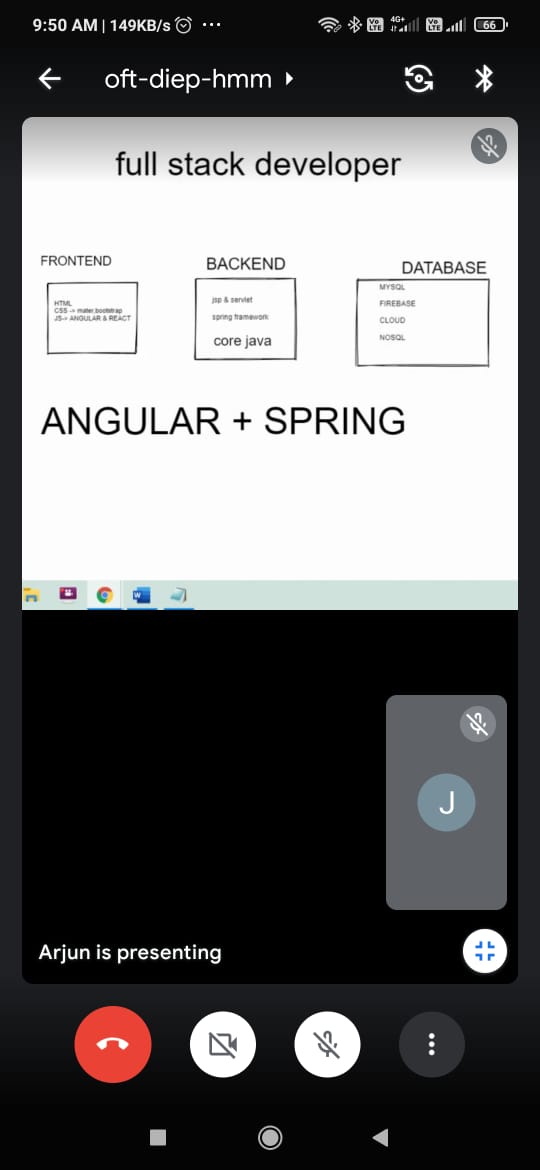
**4.Threading-, Multi Threading.=** Processing of 2 programs at a single time.

**5.Sturcture of JDBC:**

**6.API:** API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you’re using an API**.(WAITER)**

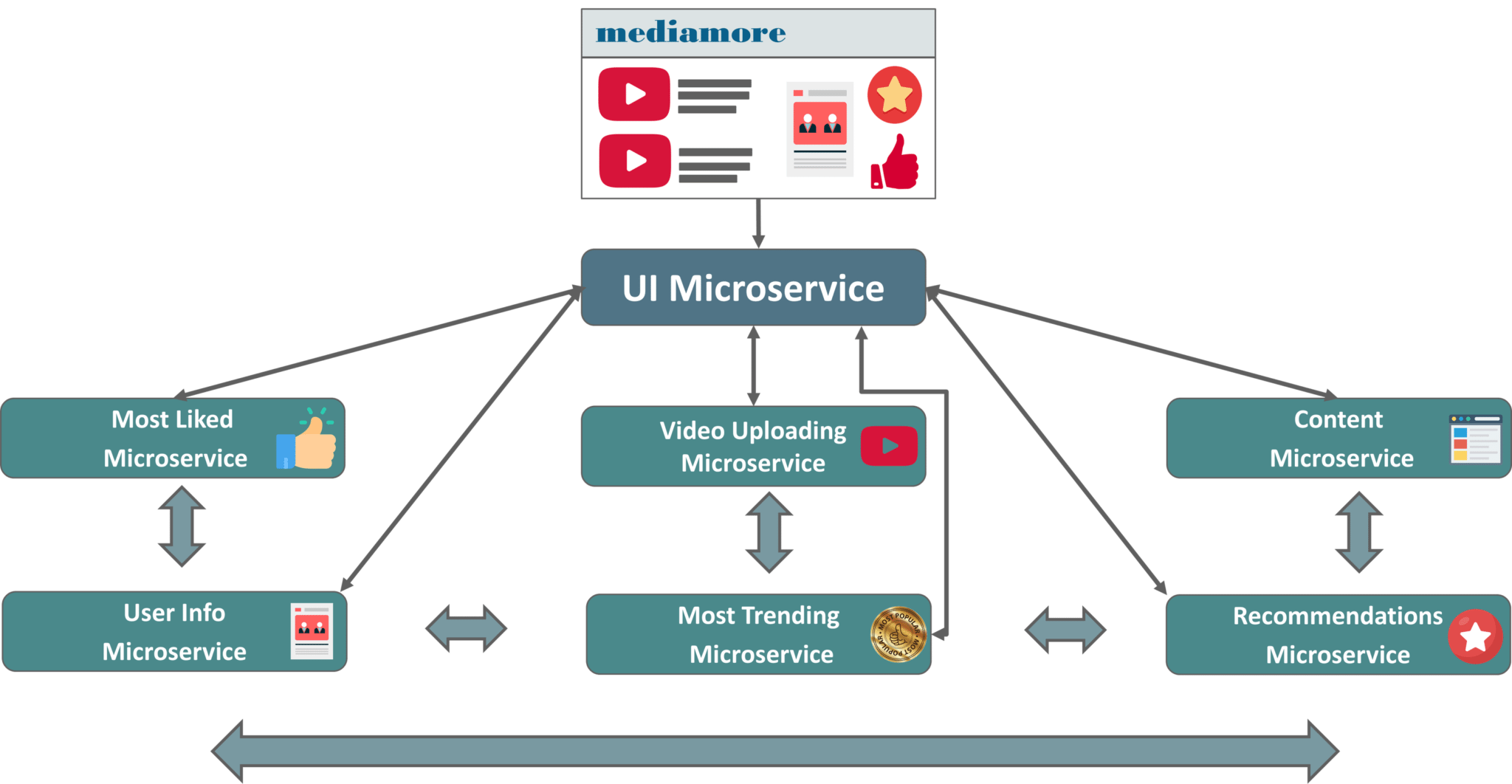
**Interface:** Boundary where two or more separate components share information or data.

We can run program without main method.



**7.MICRO SERVICES:**

* **Microservices are an architectural style that develops a single application as a set of small services. Each service runs in its own process.**
* **The services communicate with clients, and often each other, using lightweight protocols, often over messaging or HTTP.**



Here we can see that each special task is handled by small services or microservices.Here every service interacts with each other. These all microservices respond to one application or UI of YouTube

**Benefits:.**

* Each of the involved programs is independently versioned, executed, and scaled.
* These microservices can interact with other microservices and can have unique URLs or names while being always available and consistent even when failures are experienced.
* Developers can use any programming language that they’re most familiar with. This helps them come work faster, with lower costs and fewer bugs. So one application with different language microservices.
* Since your teams are working on smaller applications and more focused problem domains,. They can iterate faster, address new features on a shorter schedule, and turn around bug fixes almost immediately. They often find more opportunities to reuse code, also.
* Microservices Frameworks for Java:
* 1)Spring Boot 2)Jersey 3)Swagger