Apache thrift is a high performance, cross language RPC framework.
Allows programs written in different programming languages to talk seamlessly.
Thrift handles both Serialization and service communication.

* IDL - Interface Definition Language.
  Thrift defines data structures and service interfaces in a language neutral .thrift file.

  Example-

```
struct User {
    1: i32 id,
    2: string name
}
service UserService {
    User getUserById(1: i32 id),
    void addUser(1: User user)
}
```

  ** Supports primitive types, Containers (list, map, set), exceptions

* Code Generation - Generates code for multiple languages from thrift file.

* Protocols - Allows to define protocols for how data is serialized

TBinaryProtocol	Efficient Binary format
TCompactProtocol	More Compact Binary format
TJSONProtocol	JSON Serialization

* Transports - Allows to define how data is sent over network.

TMemoryBuffer	In memory buffer
TSocket	Blocking TCP Socket
TFramedTransport	Non blocking I/O
TFileTransport	Read/writes from files

* ServerTypes - Supports different server models.

TSimpleServer	Single threaded Blocking
TThreadedServer	Multi threaded Blocking
TThreadPoolServer	Thread Pool for handling requests
TNonblockingServer	Non blocking.

Thrift	REST	gRPC
Custom Binary Protocols	HTTP 1.1, text Based (JSON/xml)	HTTP/2
Binary Compact	JSON/XML (text Based)	Protocol Buffers
Strong Type Safety	Weak (Runtime errors possible) Ex- age: "test" REST doesn't enforce age to be integer when client reads its an error	Strong
High Performance	Low (text-based)	Very High (Binary + HTTP/2)

** Use REST for end users systems, Thrift & gRPC for internal systems for high performance. gRPC solves drawbacks of thrift like streaming

** Thrift Binary doesn't hold field names rather id and type which is the main reason for compact structure.

** Browsers doesn't natively support thrift.