Question 3
Services Interaction

1)

There are multiple ways to make two services communicate with each other.

- REST, HTTP and JSON (Synchronous)
- SOAP, Message Queuing e.g. RabbitMQ, Apache Kafka, Redis (Asynchronous)
- Websockets (Action cable in Rails)

2)

REST

PROS	**CONS**
Easy to understand,simple, flexible and fast.	Lack of state, usually making the client application heavy and difficult to maintain.
High load can be managed with help out of HTTP proxy server & cache.	Not good for confidential and very secure applications as it does not impose high level security.
Standard protection using OAuth for verification of REST requests.	Limited transfer protocol support i.e. HTTP.

SOAP

PROS	CONS
Separates the encoding and communications protocol from the runtime environment.	Usually very slow as compared to others because it uses a detailed XML format.
Built in error handling and legacy system support.	Limited to polling and to event notifications.
It is highly secure as it uses WS-security for transmission.	Firewall latency arises as firewall analyses the HTTP transport.

WEB SOCKETS

PROS	CONS
Faster connections at fewer overheads.	Doesn’t work if the browser isn’t fully HTML 5 compliant
Request/Response streaming in real time.	Doesn’t support edge caching.
End-End support for duplex connections.	Gets complex when huge dynamic interaction isn’t necessary.

3)

- **REST** should be used when the need for caching a large number of queries arises. Moreover, it is much preferred when developing light and fast apps with limited bandwidth and resources.
  This is primarily used when performance is more prioritised than security.
  Many social media platforms like Twitter are using REST for their APIs

- **SOAP** is mostly used when security is the top priority. It develops secure applications because of its asynchronous processing and subsequent invocation. Moreover, it provides WS* structure which enables stateful operations.
  This is used mostly in FinTech (banking solutions) where security cannot be compromised. A relevant example of SOAP today is Paypal's API.
- **WEB SOCKETS** provides the fastest communication and fewest overheads, so it is really good for applications which require continuous real-time data delivery. As the connection is kept open for a long time, sockets speed up the data transmission and real-time fetching. This is usually used for Chat and Gaming apps which demand continuous and quick data exchange. A prominent example of web sockets usage is WhatsApp. They use sockets for swift message delivery.