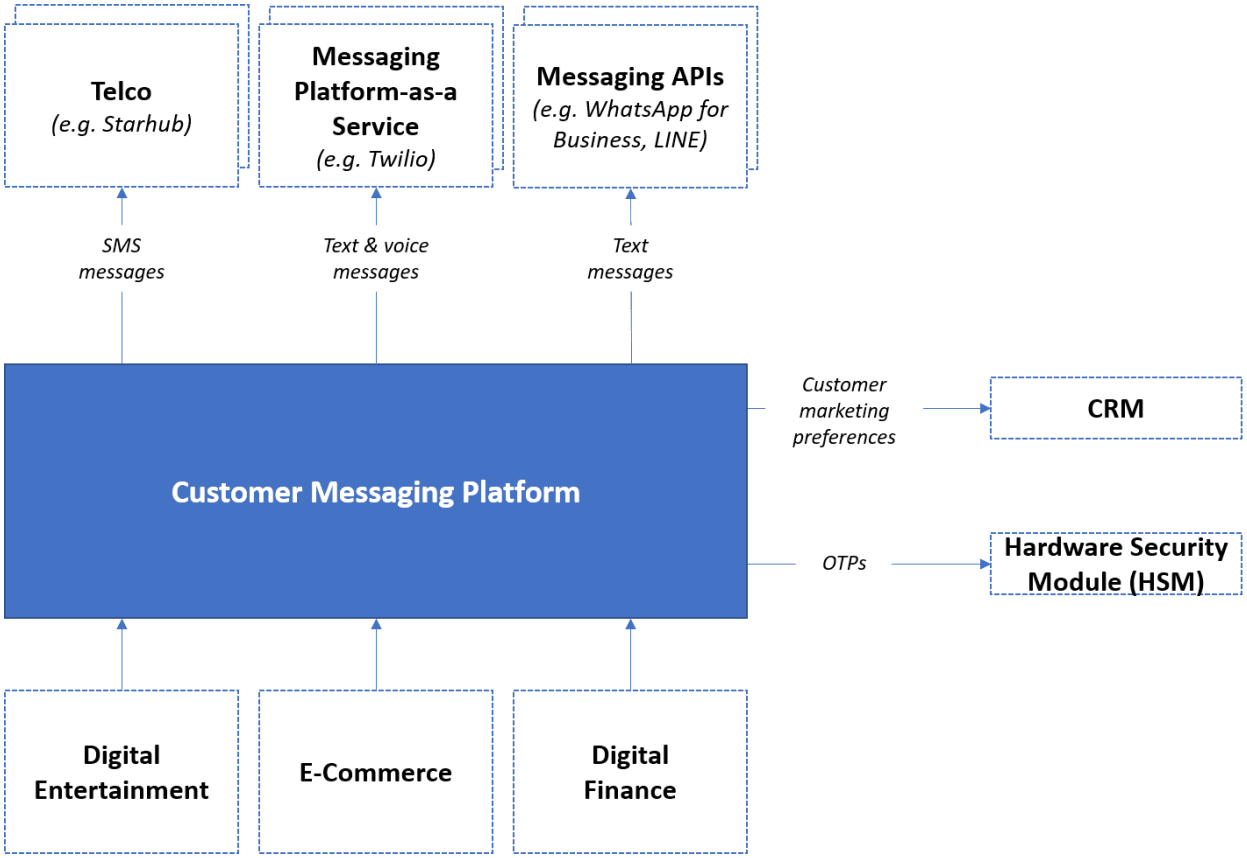
BRIEF

You meet with an Internet company ("The Client") that develops and operates many different mobile applications, including mobile games, e-commerce and digital financial services. The Client has hundreds of millions of customers globally, and are the market leader in many of the businesses they operate. This is supported by a whole suite of technology and platforms that have been developed within the company.

SCOPE

The Client is interested in appointing a team to build a new communications platform to centralise all outbound communications to their customers. The goal is to have a single platform that the Client's developers can use to programmatically send messages from their service to customers. Example of use cases for this system are sending of one-time passwords (OTP), order updates, payment notifications, marketing messages, order delivery alerts etc. Messages may be delivered in the form of emails, SMSes, voice messages, or even mobile push notifications. The source for these systems can be any system or service managed by the Client. You should assume that all interfaces between internal and external systems are via APIs (HTTP/HTTPS).



These are a few mandatory system functions that the Client is expecting:

- **The system should deliver messages as soon as possible, but not earlier or later than instructed**. For example, an OTP that is valid for only 60 seconds should be delivered as soon as requested, while a message that is scheduled to be sent at midnight should only be sent at that time.
- **The system should be able to handle many concurrent requests**. You may assume that at the start there are hundreds of services that will use this system, and that there are tens of millions of requests per day. There may be hundreds of thousands of messages to be sent in a single request. However this may scale to thousands of sources and billions of requests per day.
- **The system should use the appropriate delivery method and gateway based on business rules.** For example, SMS messages may be routed to different external gateways based on customers' mobile prefix (e.g. numbers starting with +65123 are sent to Telco A, while +65456 are sent to Telco B). Or System A may only send messages through Gateway X.
- **The system should be built to handle failures gracefully**. For example the system should re-try or use an alternate delivery method automatically if an external delivery gateway does not respond within a specific time. Another example would be handling sudden increase in requests during seasonal peaks.
- **The system should only be used by authorized source systems, and only for permitted use cases**. For example, System A may send order update notifications, but not used for sending OTPs.
- **The system should only send messages to legitimate recipients**. For example, it should reject invalid recipients (e.g. incomplete or inaccurate records, blacklisted numbers or regions etc). Another example: all systems are not allowed to send marketing messages to customers who have opted out of marketing communications (as indicated in an external marketing database), but non-marketing messages (such as OTP) may be sent.
- **The system should have mechanisms to prevent abuse/inappropriate content**. This could be in the form of matching message content with approved templates, keyword matching, or any other techniques that meet the objective. A system should only be allowed to send a certain number of messages in a given time interval; similarly a customer should not receive too many messages in a given timeframe.
- **The system should log and track every message that has been sent for audit and compliance purposes**. The system should allow the administrator to quickly and efficiently find historical records of past messages.
- **The system should account for every message sent and provide accurate reports**. This is for both operations and business teams to monitor message volume and costs on a adhoc, daily, weekly and monthly basis.

OBJECTIVE

The client has requested your team to design and build a working prototype of this system within 4 weeks.