Messaging System Architecture

1. Scope of the Document

The goal of this document is to provide a high level instant messaging backend architecture. Remarks in terms of scalability, performance will be discussed in a separate session.

2. High Level Architecture

There are 3 main aspects that should be part of an Instant Messaging system and/or should be taken into account during its implementation phase:

- Rest API
- Caching Subscriber/Publisher Storage
- Authorisation and Authentication Provider

Rest API

Rest API endpoints should be provided to access the messaging application either programmatically (via SDK) or via a web/mobile interface. Any Rest API is meant to evolve over time and the approach taken in versioning it is crucial to avoid large scale impact when moving to a new version. This is particularly relevant when dealing with both B2C and B2B customers.

Storage

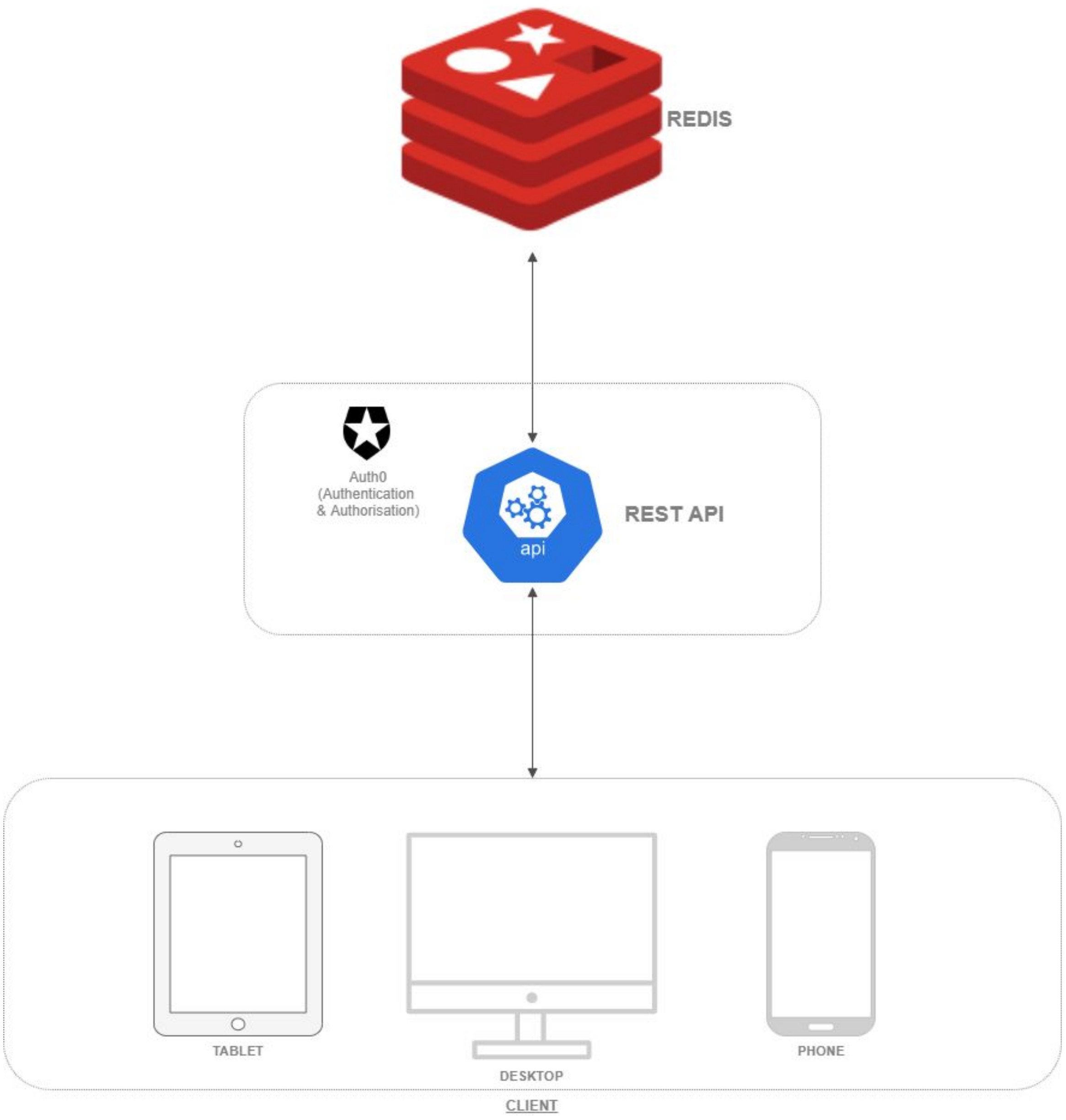
Nowadays, there are various storage solutions from SQL/Relational Databases to NOSQL and even Graph databases (such as Titan or Neo4j). Because of the nature of the transactions and the high volume of the incoming data, a NoSQL solution should be preferred. In the realm of NoSQL Databases, REDIS (Remote Dictionary Server) stands out by bringing the following features:

- It is an in-memory data store, which means that reading times are generally better than access disk Databases. Because of that, REDIS is popular and sought after for session management, real time analytics, messaging/chat systems, media streaming etc.
- Unlike key-data stores that normally offers limited data structures, REDIS offers a variety of data structures (e.g. Hashes)
- REDIS follows a primary-replica architecture both on single node or clustered topology and it allows building highly available solutions
- REDIS supports Pub/Sub architecture and allows setting up queues in a much simpler way than, for example, RabbitMQ.

Authentication/Authorisation Provider

Another important element in an instant messaging architecture is security in terms of identity management, user authentication and authorisation. Auth0, for example, offers a fully fledged IAM Solution and Enterprise features such as SSO and Social Logins.

3. Diagram



4. Remarks

Scalability and Performance are the key metrics to watch out for when building and deploying an instant messaging service. In order to have a better control and management of such service, it is important to choose the right solution in terms of container orchestrator (e.g. Kubernetes) and to dedicate time to achieve the ad hoc configurations when it comes to CPU, mounted volume per each pod, service topology, load balancing etc. To be able to detect issues with performances, dashboard and monitoring solutions such Grafana or SpringBoot Actuator + Prometheus can give very valuable insights on the service’s health.