

Design Notification Service System.



Notification Service System: (Scalable)

Functional Requirements:

- 1) Send Notifications
- 2) pluggable (open for extension
Say add whatsapp Notifications)
- 3) SaaS (Software as a service so that can be used internally or externally)
- 4) prioritization (certain messages have certain priority)

Non-Function Requirements: 5) Bulk Notification

- 1) Scalability
- 2) High Availability

List of Services:

- | | |
|---------------------------|---------------------|
| 1) Notification Service | 8) Email Handler |
| 2) Notification Validator | 9) SMS Handler |
| 3) Prioritizer | 10) in-APP Handler |
| 4) Notification Handler | 11) User Preference |
| 5) User Service | |
| 6) Rate Limiter | |
| 7) Request Counter | |

List of Databases:

1) Redis or Memcached : (in-memory DB)

It is used near Rate limiter to check if threshold is met or not

User-id	Count
143	2
152	1

2) Cassandra DataBase: (wide-column DB)

For auditing purpose, a wide-column DB Cassandra is used at Notification Handler to keep track of notifications sent

User-id	Type of Notification	Client-id
1	Email	2
3	Sms	2

3) User Preference DB: (RDBMS):

Some users prefer Email notification while some don't like notifications. To keep

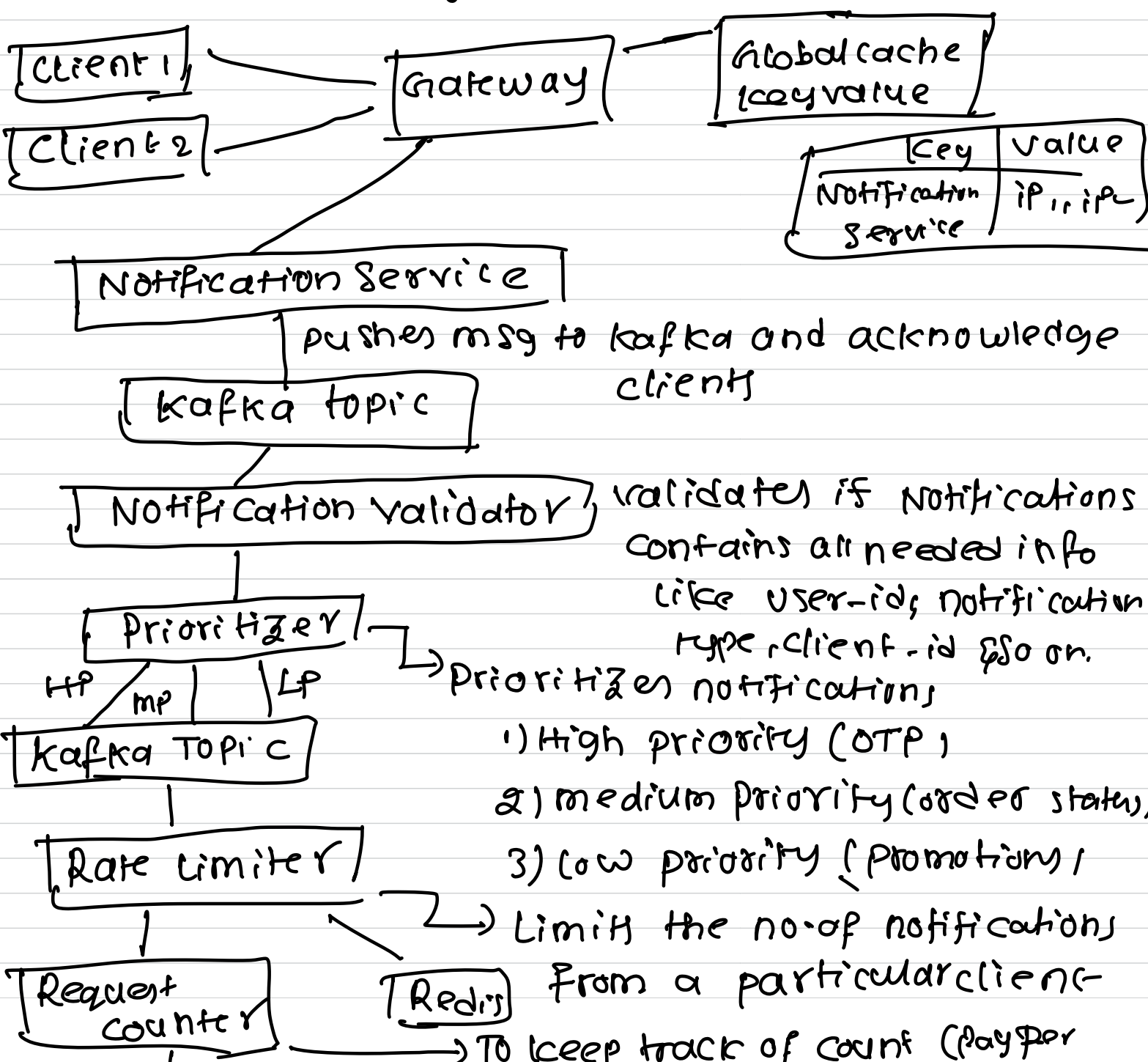
track all the preferences we need DB

User-id	Preference
1	Email
2	No

4) User Transaction Data: (Time-based DB)

user-id	product	time stamp
1	milk	3 days ago
2	milk	2 days ago

user transaction helps in selecting users while performing Bulk Notification



request

Provides data
related to
user
preference

User
Preference

mysql
USER PREFERENCE
DB

Notification Handler

→ Segregates based
on notification type
& pushes into respective
notification handlers
Based on user preference

Kafka topic

Email Handler

USER
Service

Redis

Provides the
information related
to no. of notifications
received by the user
till time

Sms Handler

INAPP Handler

Firebase
Cloud Service

→ Apple Push Notification Service

Add whatsapp if needed

