



Google Cloud

Messaging

Francesco Nerieri



# Google Cloud Messaging

400% growth in 10 months





# Google Cloud Messaging



400% growth in 10 months





# Google Cloud Messaging



400% growth in 10 months



200K+ QPS





# Google Cloud Messaging



400% growth in 10 months



200K+ QPS



60+% of most popular apps





# Google Cloud Messaging

Surf  
Server

GCM

Surf  
App



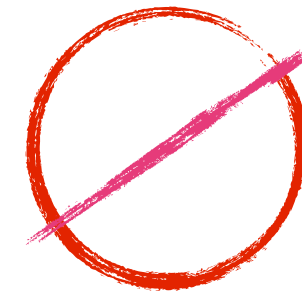


# Google Cloud Messaging

Surf  
Server

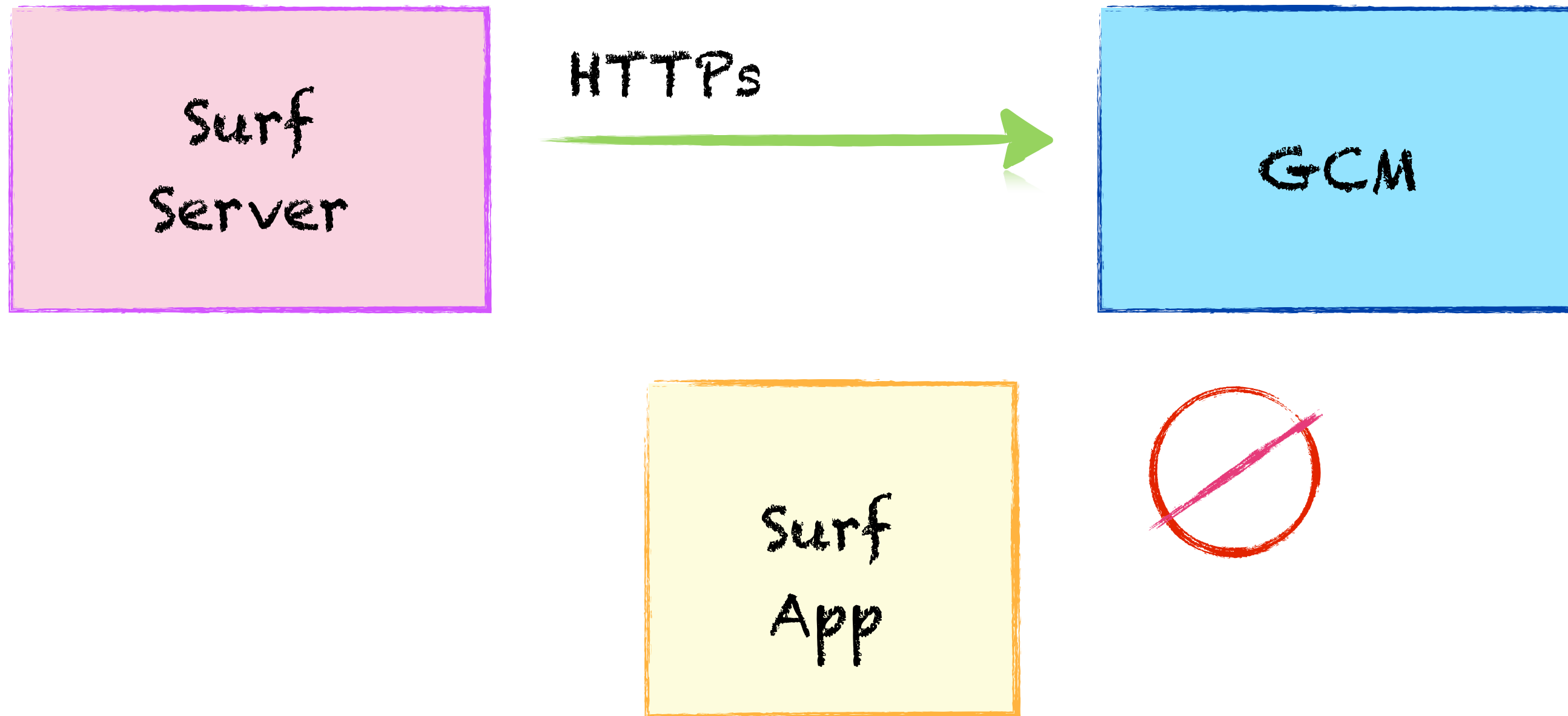
GCM

Surf  
App



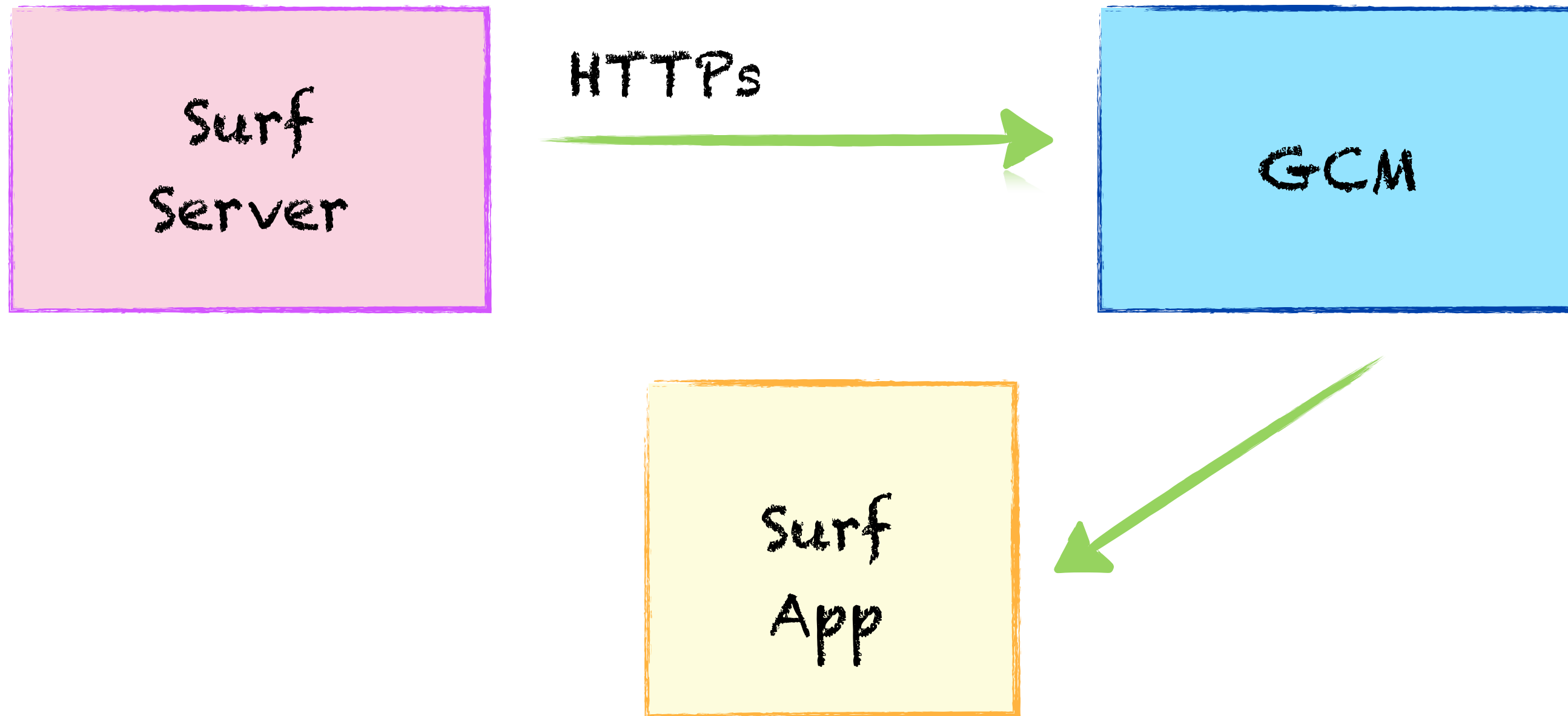


# Google Cloud Messaging





# Google Cloud Messaging



# Google Cloud Messaging *Send to Sync*





# Google Cloud Messaging *Send to Sync*

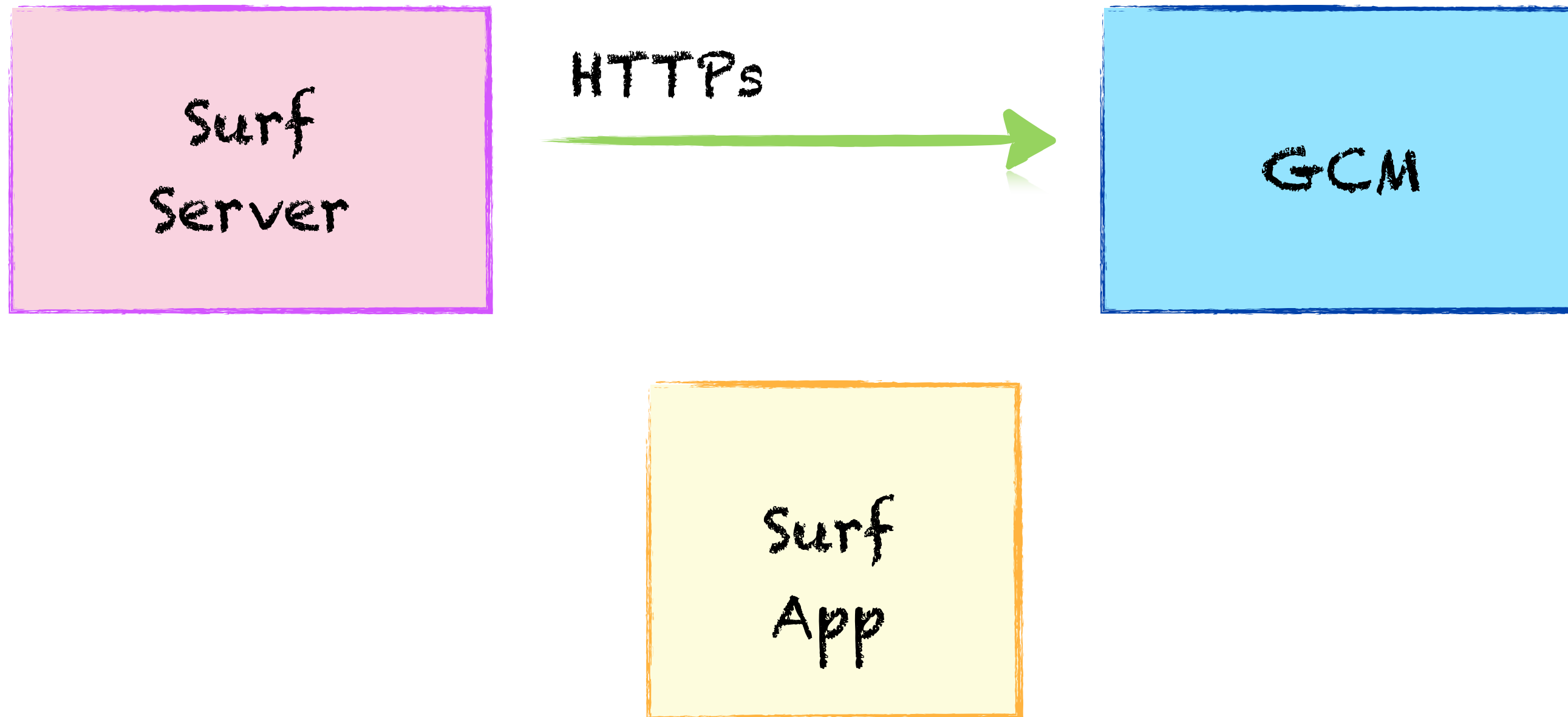
Surf  
Server

GCM

Surf  
App

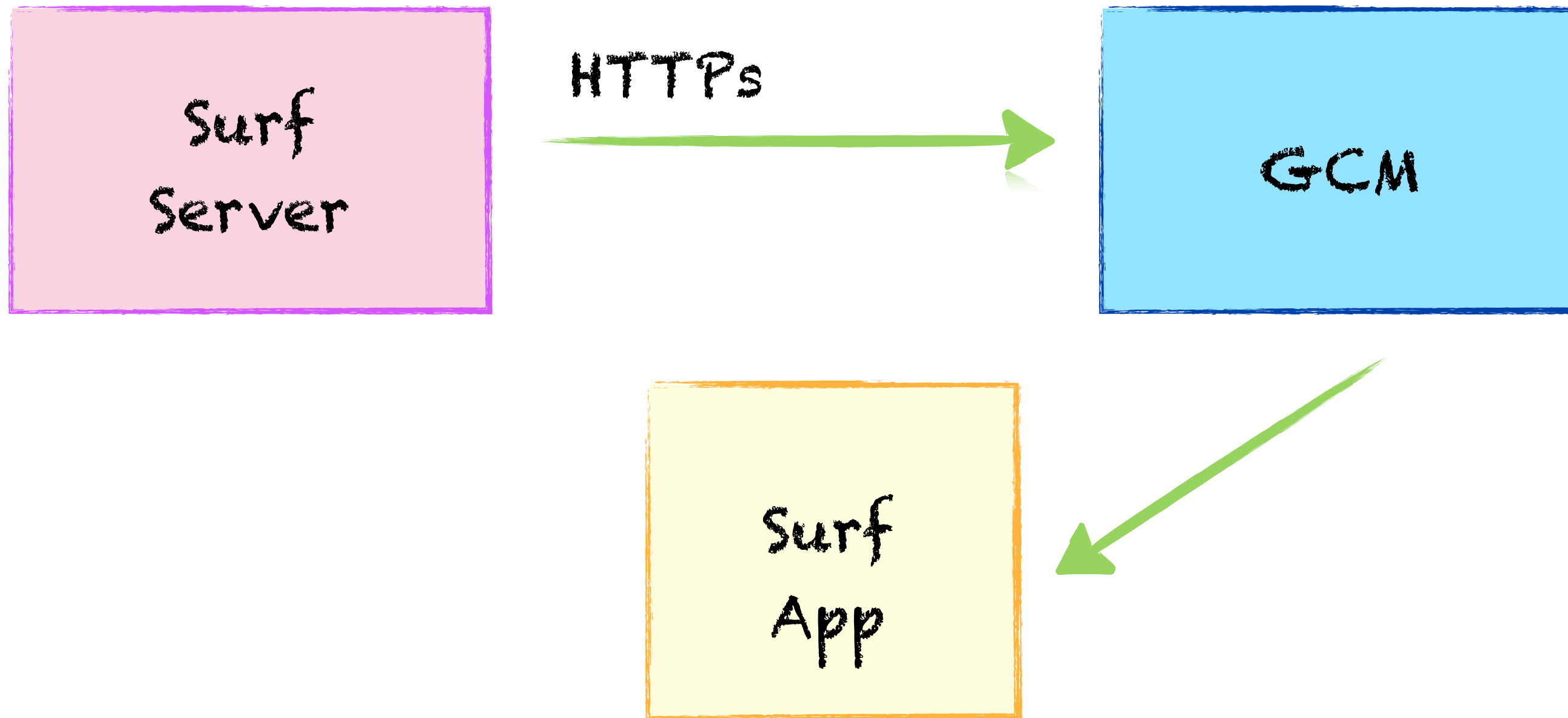


# Google Cloud Messaging *Send to Sync*

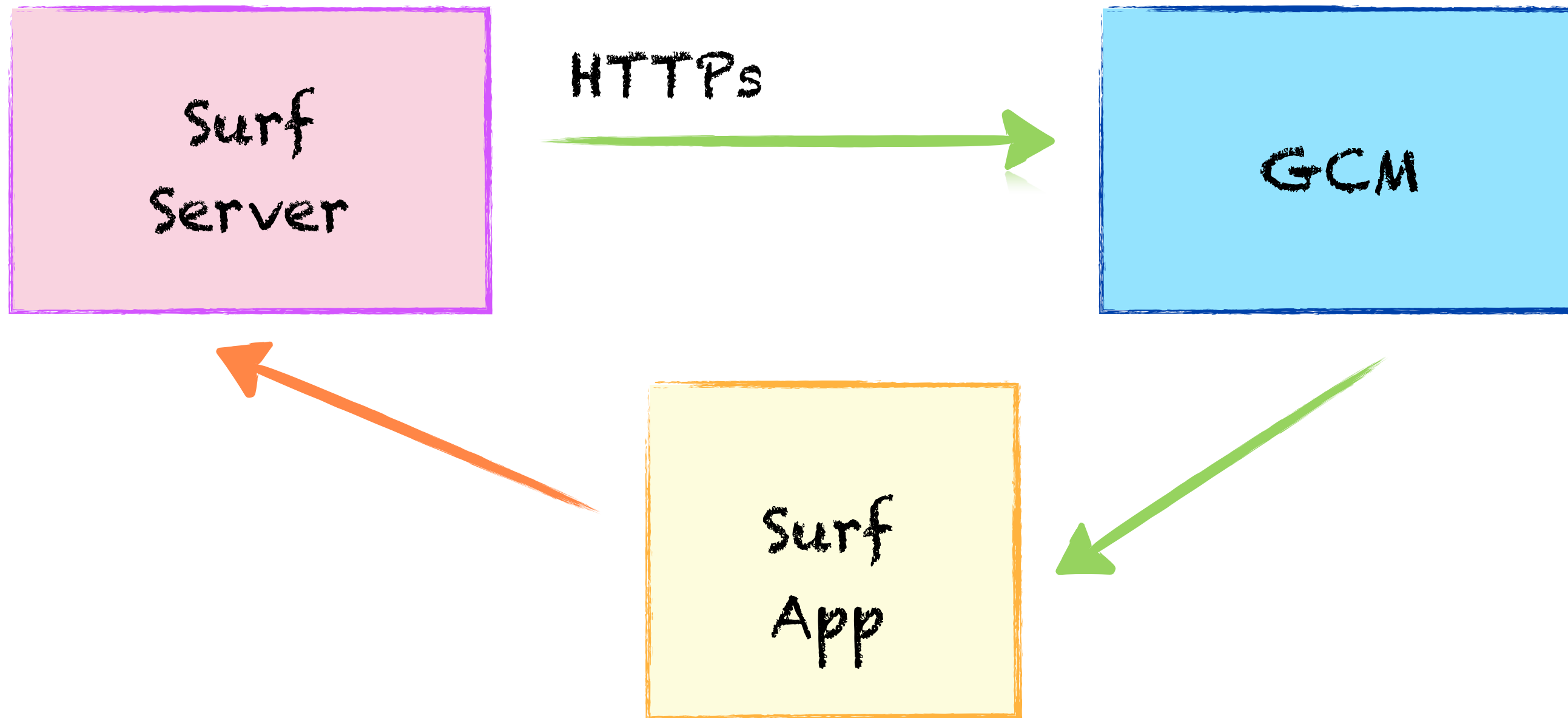




# Google Cloud Messaging *Send to Sync*

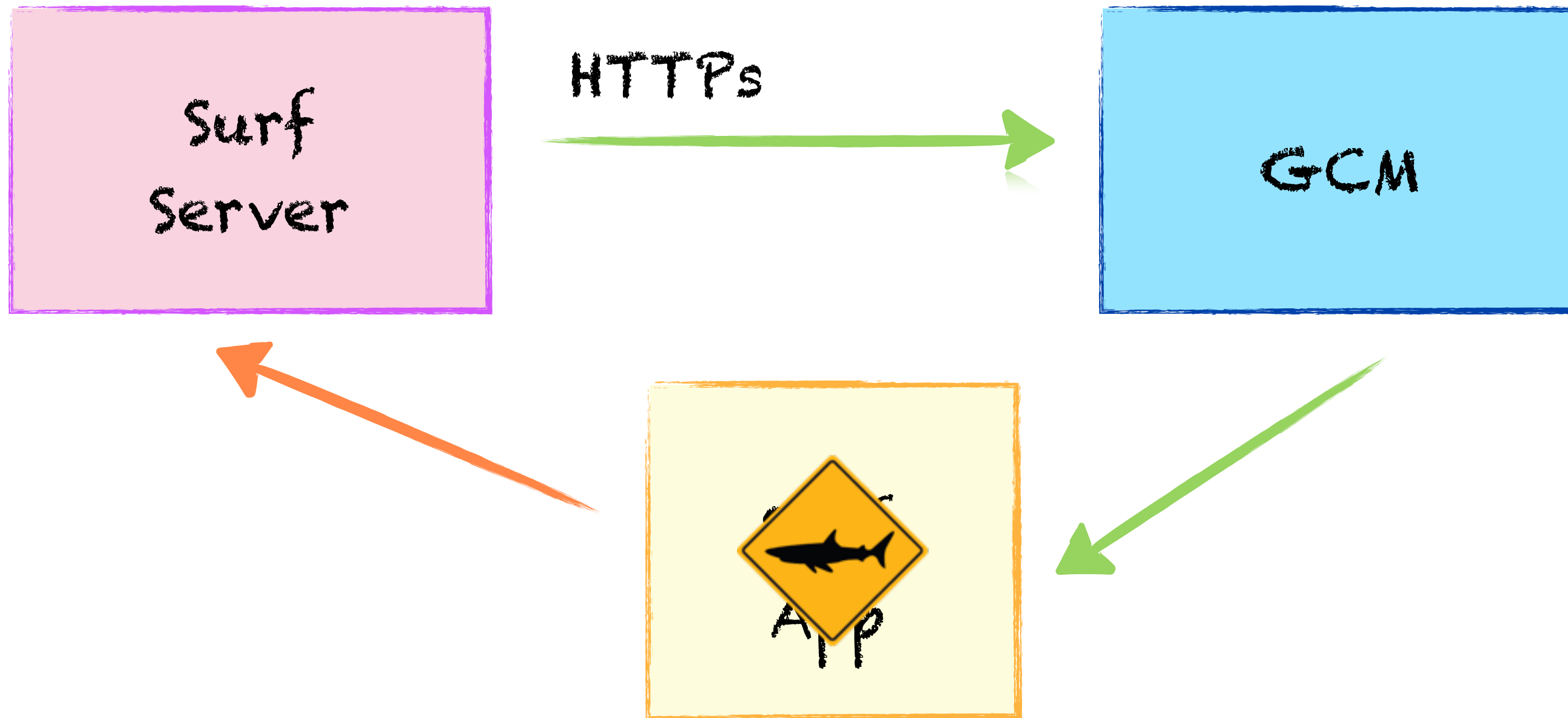


# Google Cloud Messaging *Send to Sync*





# Google Cloud Messaging *Send to Sync*



# Google Cloud Messaging *Send Data*

Surf  
Server

GCM

Surf  
App





# Google Cloud Messaging *Send Data*

Surf  
Server

GCM



# Google Cloud Messaging APIs

Time To Live and Delay While Idle



# Google Cloud Messaging APIs



Time To Live and Delay While Idle





# Google Cloud Messaging APIs



Time To Live and Delay While Idle



Multiple Senders



# Google Cloud Messaging APIs



Time To Live and Delay While Idle



Multiple Senders



Message Multicasting

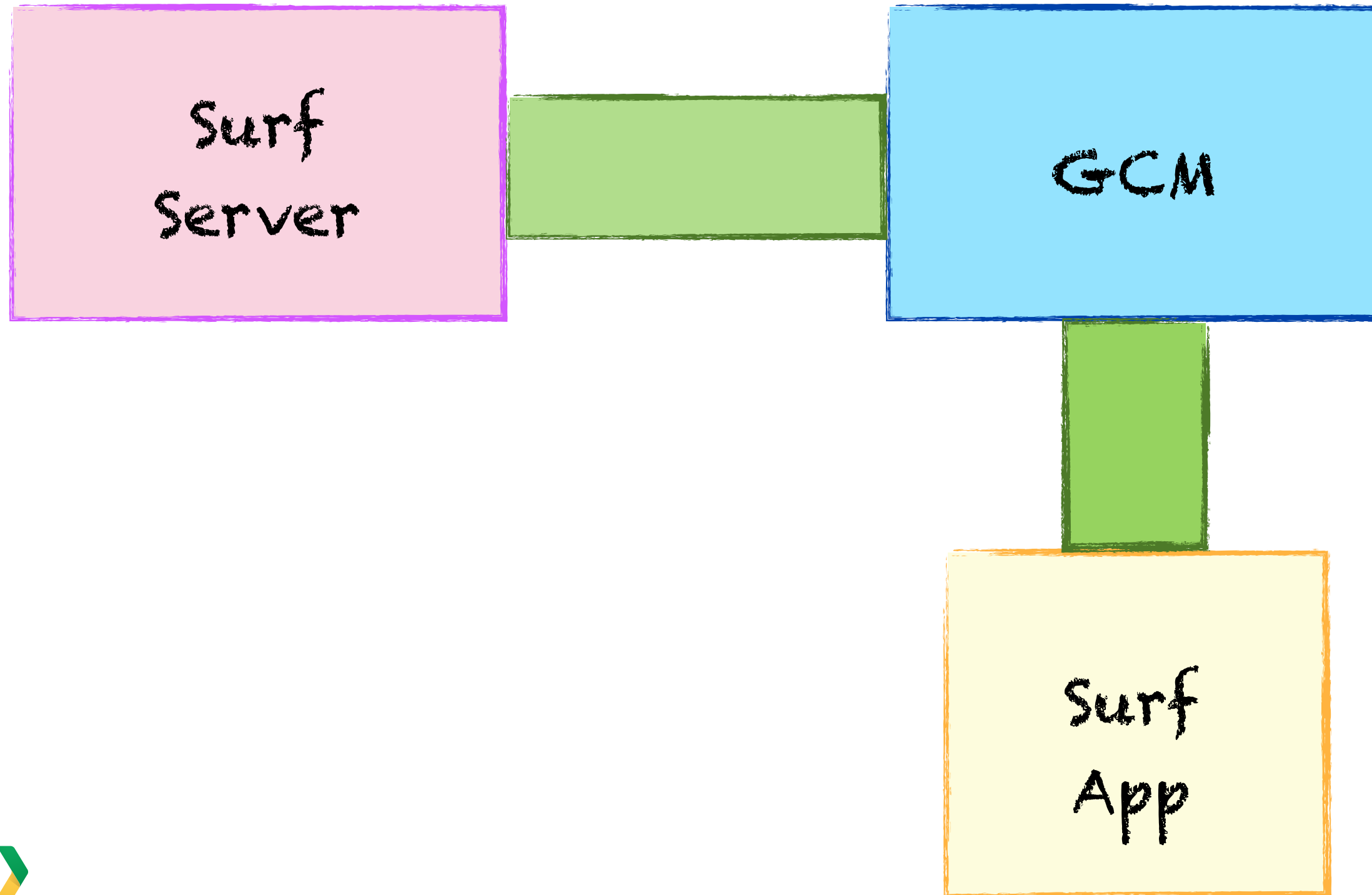




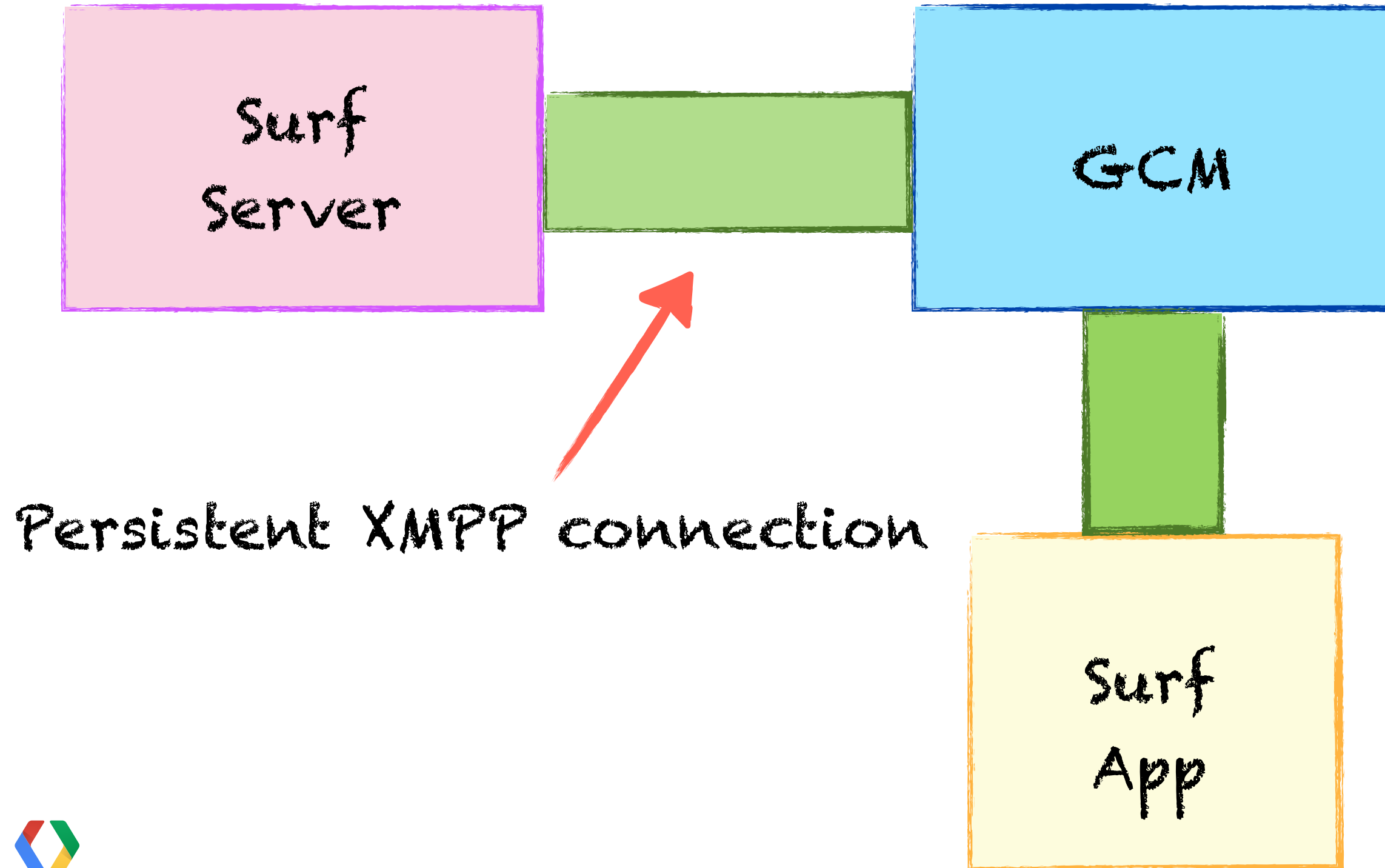




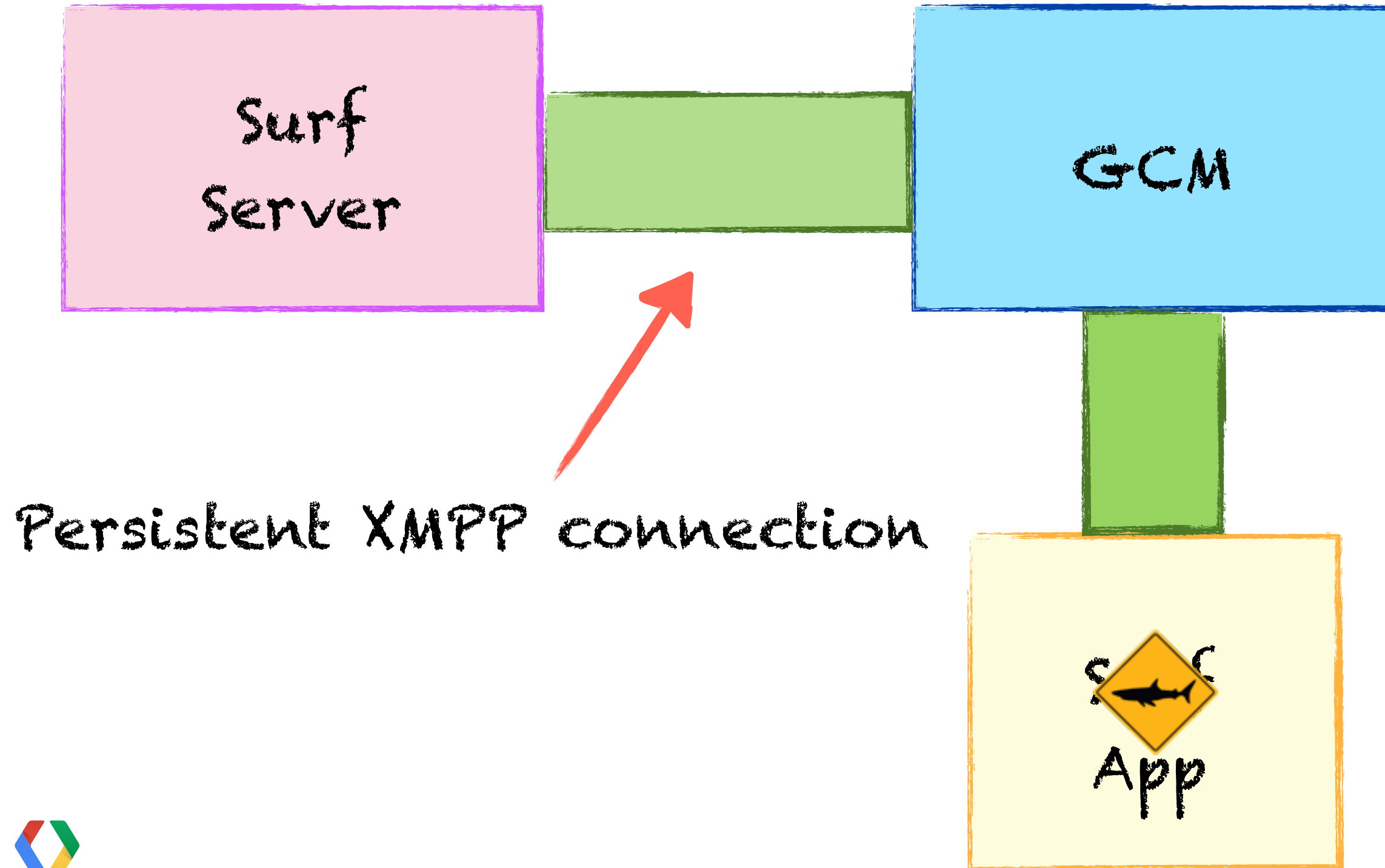
# Google Cloud Messaging XMPP Endpoint



# Google Cloud Messaging XMPP Endpoint



# Google Cloud Messaging XMPP Endpoint



```
SERVER = ('gcm.googleapis.com', 5235)
```

```
client = xmpp.Client('gcm.googleapis.com', 5235)
```

```
client.connect(server=SERVER,secure=1,use_srv=False)
```

```
client.auth(PROJECT ID, API KEY)
```





## XMPP Client:

```
<auth mechanism="PLAIN"
  xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
  MTI2MjAwMzQ3OTMzQHBy [...]
</auth>
```

## GCM Server:

```
<success xmlns="urn:ietf:params:xml:ns:xmpp-sasl"/>
```



```
node = "<message>  
  <gcm xmlns:google:mobile:data>  
    JSON payload  
  </gcm>  
</message>"
```

```
client.send(xmpp.protocol.Message(node))
```



```
{  
  "to": "REGISTRATION_ID",  
  "data": {"hello": "world", }  
  "collapse_key": "collapse"  
  "time_to_live": "600",  
  "message_id": "m-1366082849205"  
  "delay_while_idle": true/false  
}
```



# Message stream





# Message stream



ACK

```
<gcm xmlns="google:mobile:data">  
  
  { "from": "REGISTRATION_ID",  
    "message_id": "m-1366082849205"  
    "message_type": "ack" }  
  
</gcm>
```



NACK

```
<gcm xmlns="google:mobile:data">  
  
  { "from": "REGID",  
    "message_id": "m-1366082849205"  
    "error": ERROR_CODE,  
    "message_type": "nack" }  
  
</gcm>
```



# XMPP Endpoint

Up to 10 connections



# XMPP Endpoint



Up to 10 connections





# XMPP Endpoint



Up to 10 connections



Message streaming with ACKs/NACKs



# XMPP Endpoint



Up to 10 connections



Message streaming with ACKs/NACKs



Same GCM APIs, no Multicasting





A wooden sign with a blue Android robot head and the word 'Upstream' in red. The sign is mounted on a wall with a collage background of brown and tan paper. The Android robot head is blue with two antennae and two circular eyes. The word 'Upstream' is written in a red, hand-drawn font. The sign is made of horizontal wooden planks and has two circular holes on the left side. The background is a collage of brown and tan paper with some red and white patterns.

Upstream



# Google Cloud Messaging **Upstream**

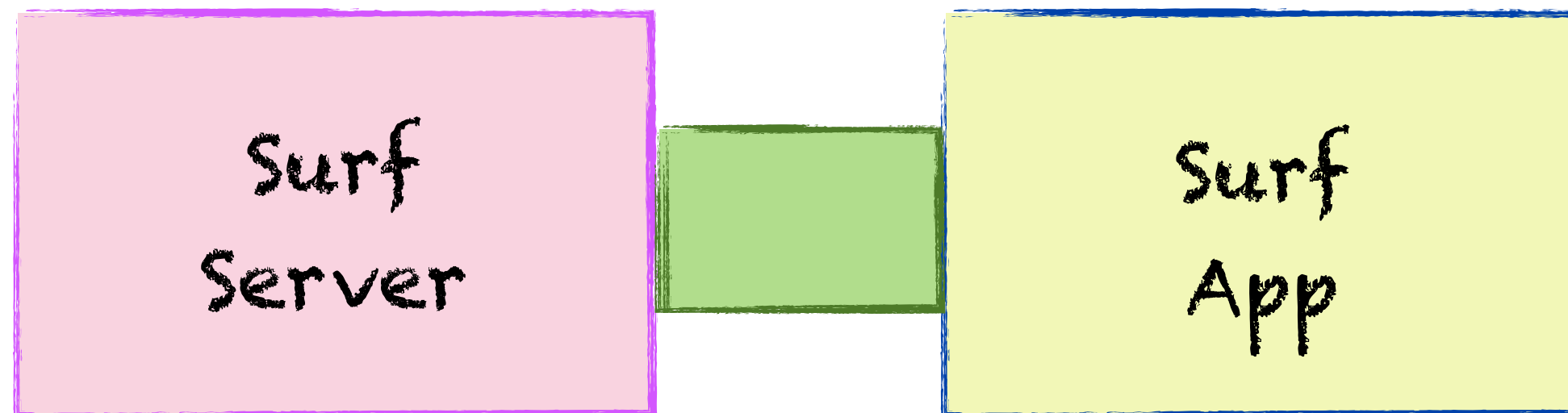


# Google Cloud Messaging **Upstream**





# Google Cloud Messaging Upstream

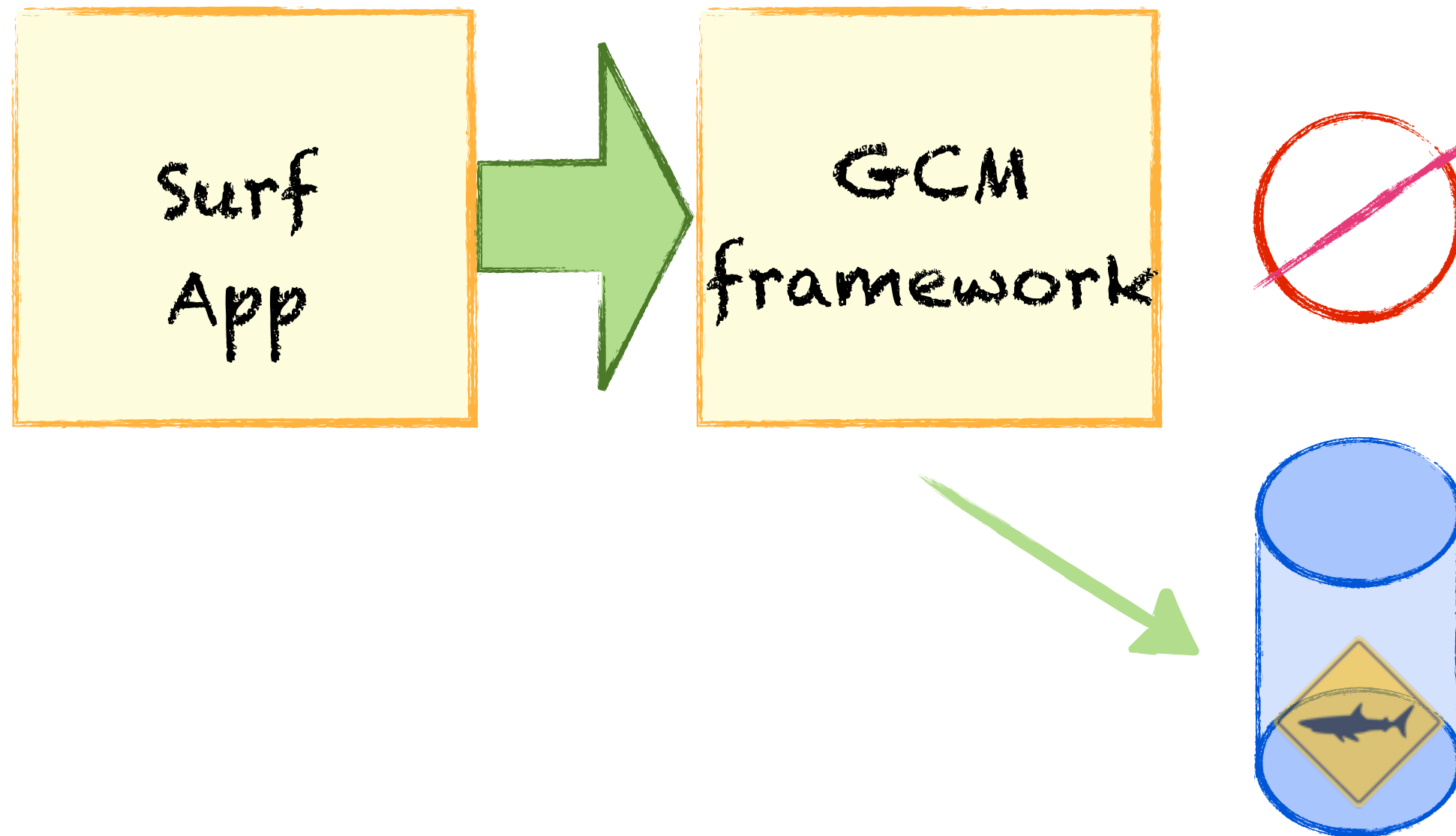


# Sending an **upstream** message

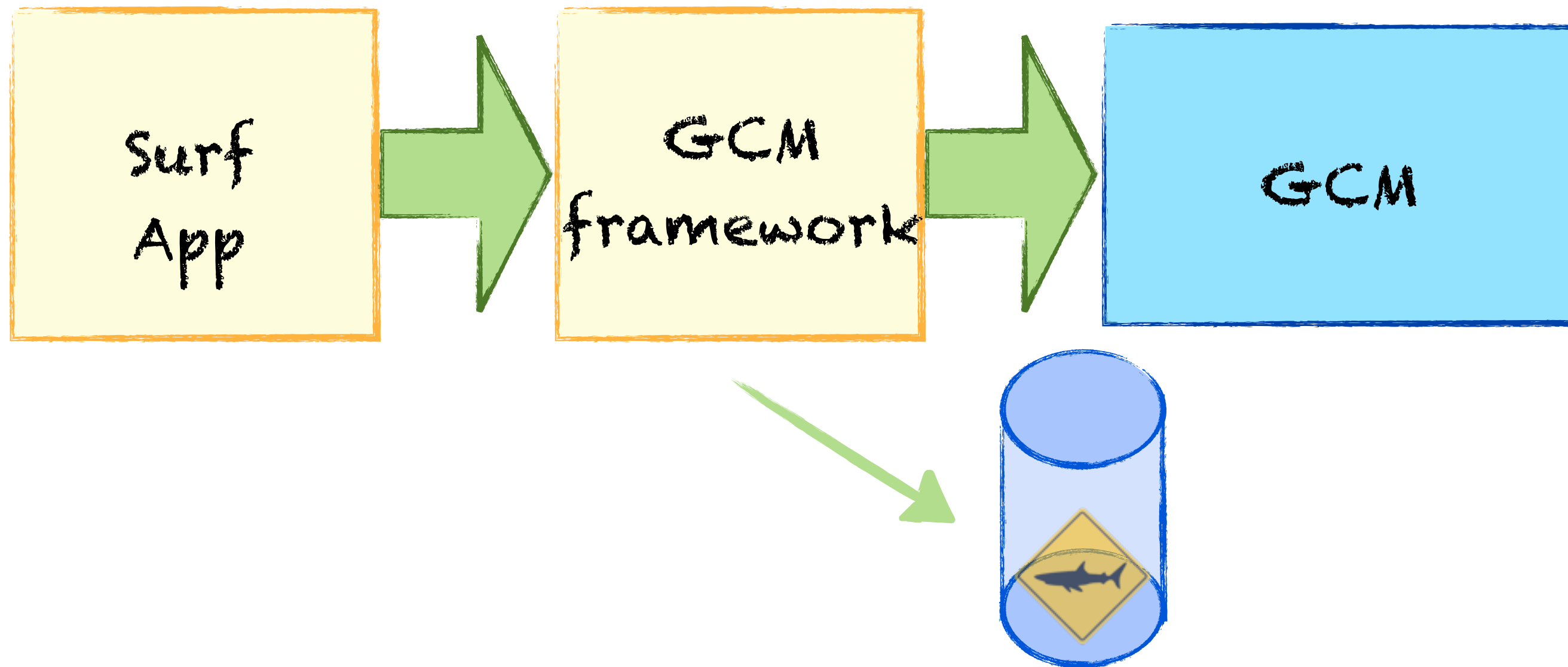
Surf  
App



# Sending an **upstream** message



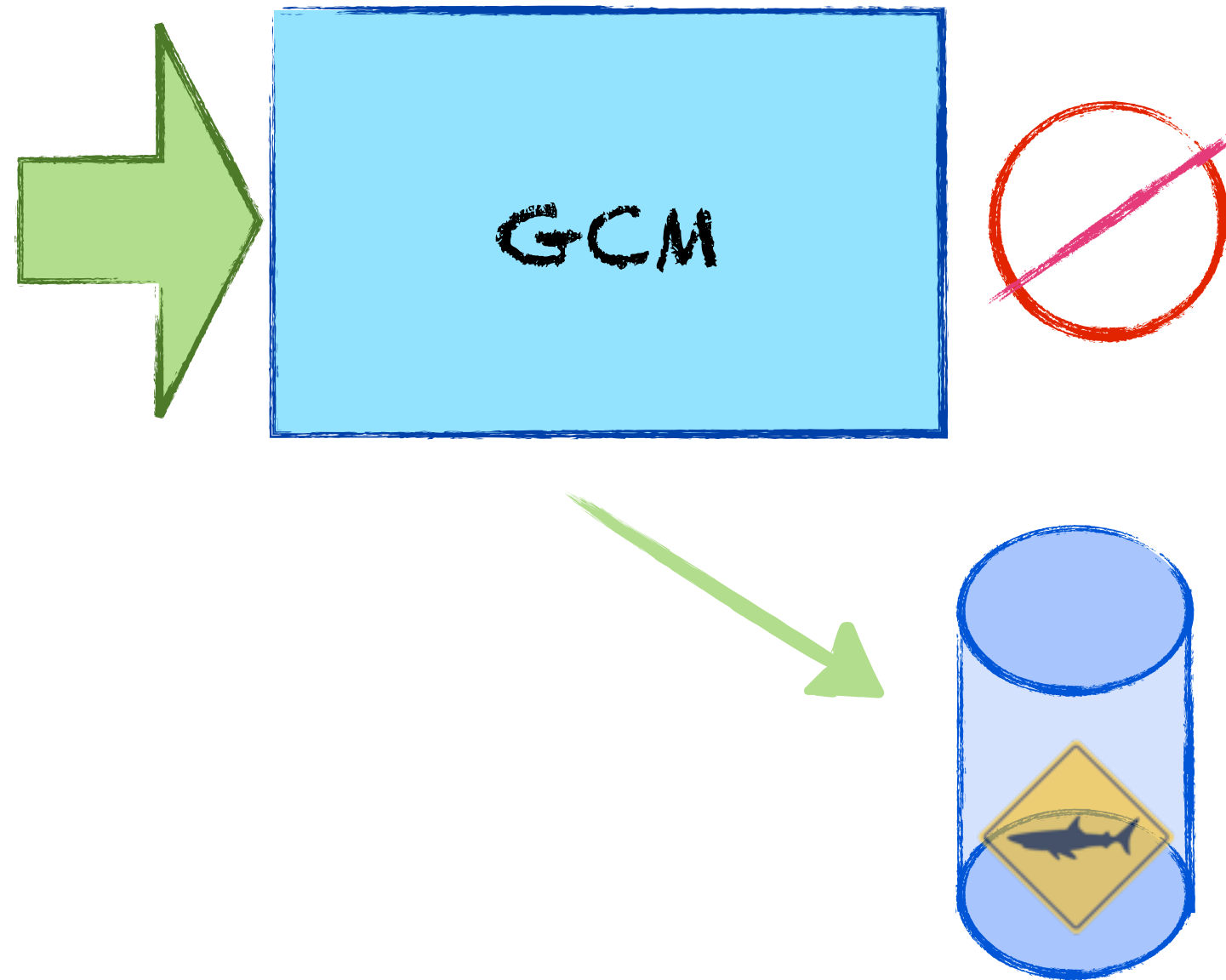
# Sending an **upstream** message



# Sending an **upstream** message

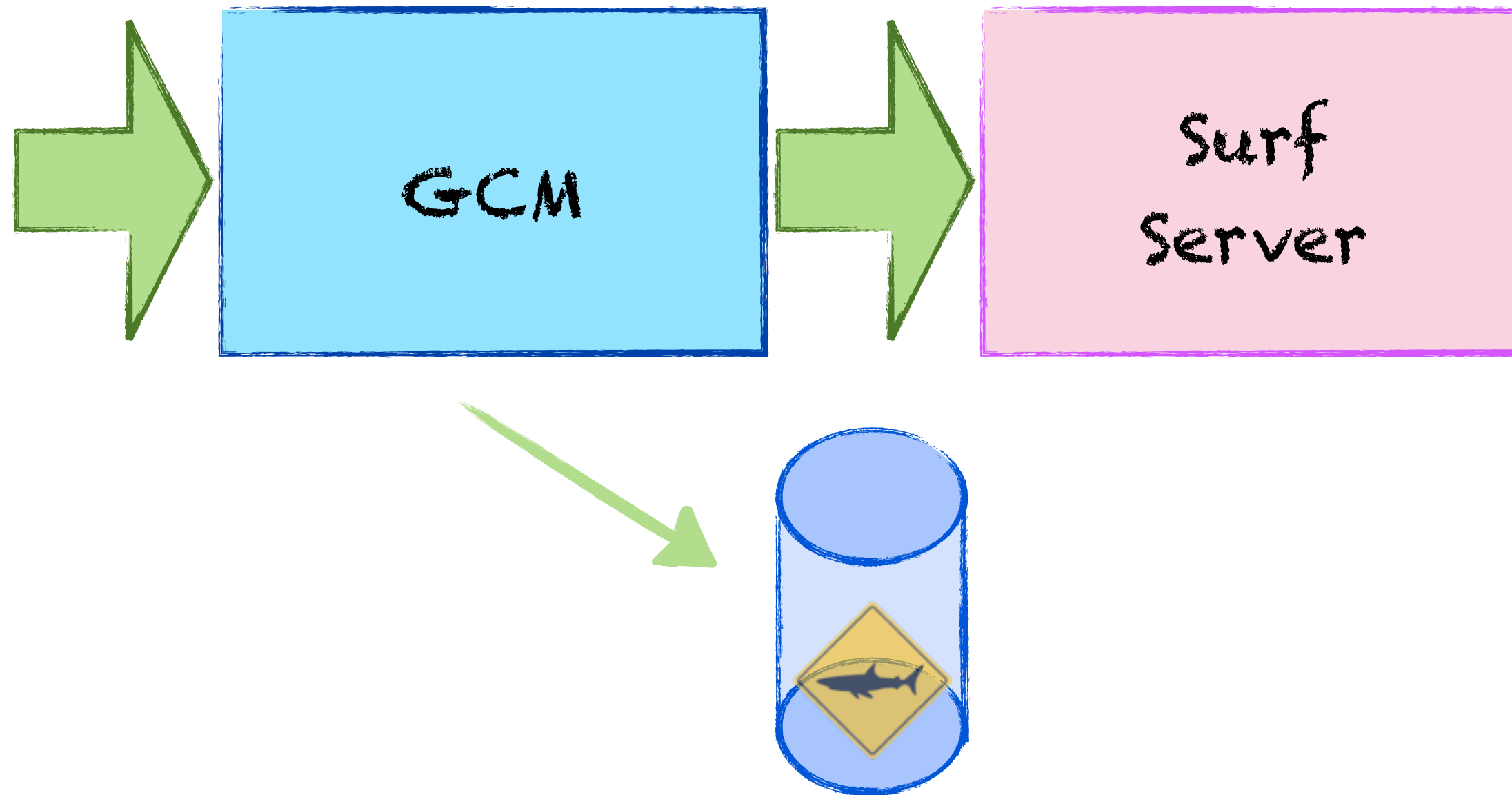


# Sending an **upstream** message





# Sending an **upstream** message



```
GoogleCloudMessaging gcm =  
    GoogleCloudMessaging.get(context);
```

```
String to = PROJECT_ID
```

```
gcm.send(to, msgId, data);
```



# Time To Live API



# Time To Live API





# Time To Live API



# Time To Live API



```
GoogleCloudMessaging gcm =  
    GoogleCloudMessaging.get(context);  
  
int timeToLive = [0 seconds, 4 weeks]  
  
gcm.send(to, msgId, timeToLive, data);
```



# Google Cloud Messaging **Error Handling**

No Connection



# Google Cloud Messaging **Error Handling**



No Connection





# Google Cloud Messaging **Error Handling**



No Connection



Message is too big



# Google Cloud Messaging **Error Handling**



No Connection



Message is too big



Too many outstanding messages



```
@Override  
public void onReceive(Context context, final Intent intent) {  
    if (MESSAGE_TYPE_SEND_ERROR.equals(messageType)) {  
        [...]  
    }  
}
```



# Message stream



# Message stream





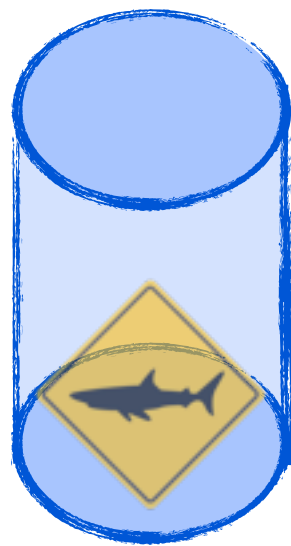
ACK

```
<gcm xmlns="google:mobile:data">  
  
  { "to": "REGID",  
    "message_id": "m-1366082849205"  
    "message_type": "ack" }  
  
</gcm>
```

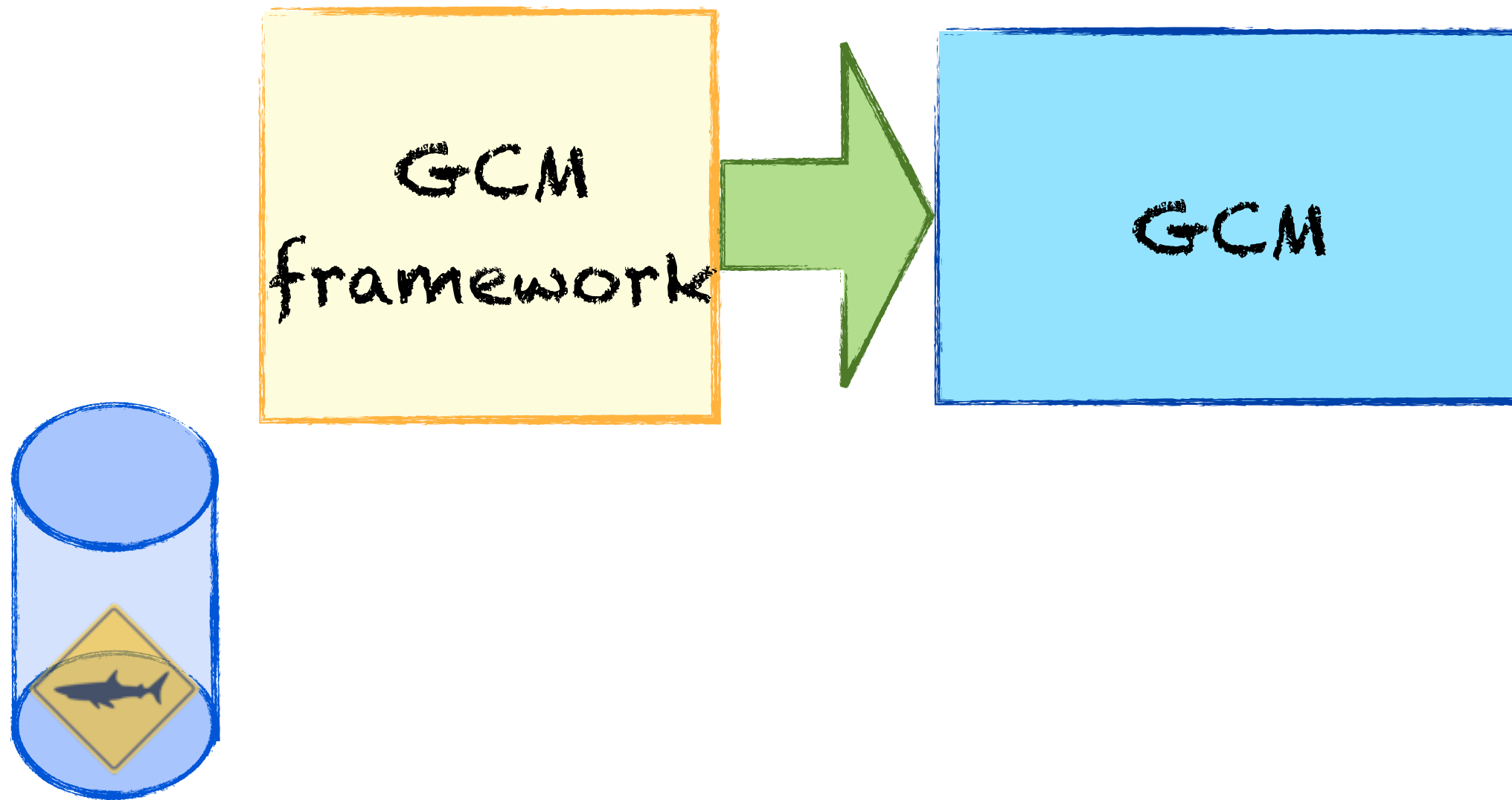


# Deleting ACK'd messages

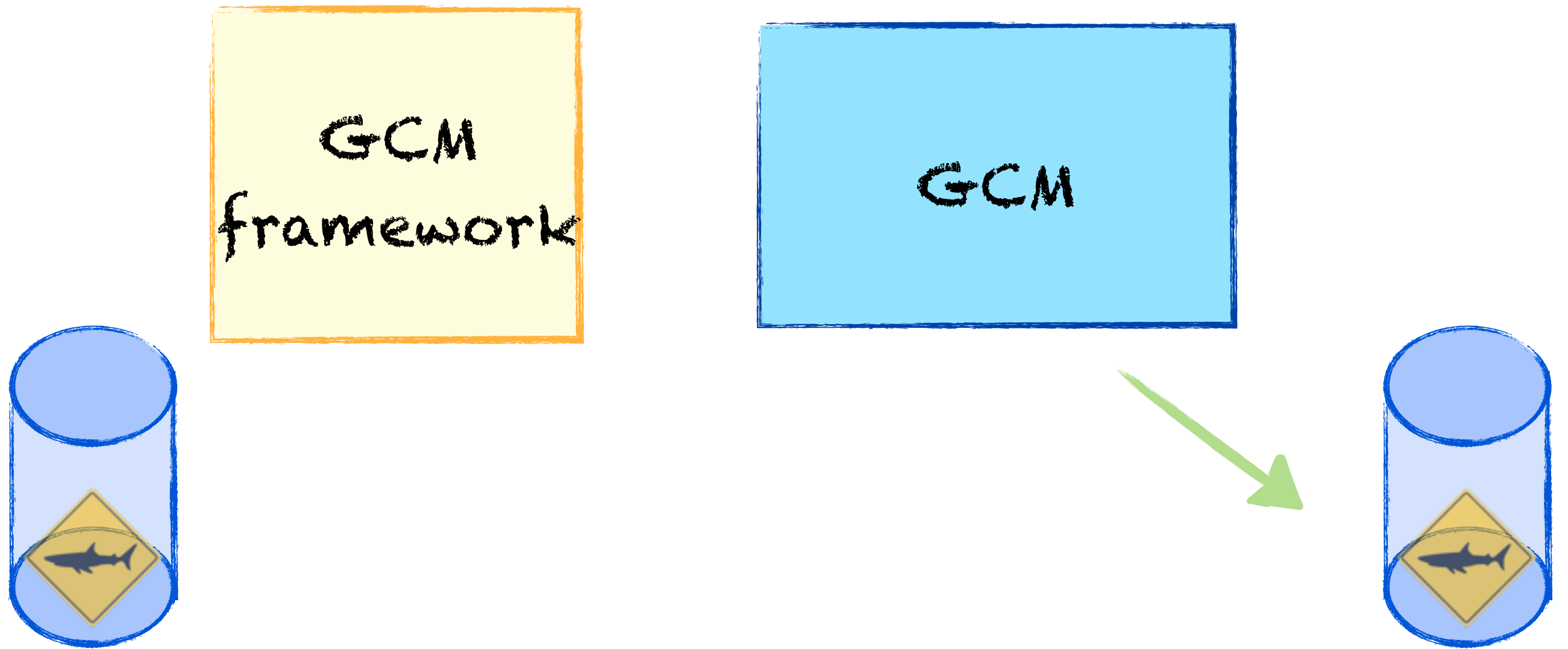
GCM  
framework



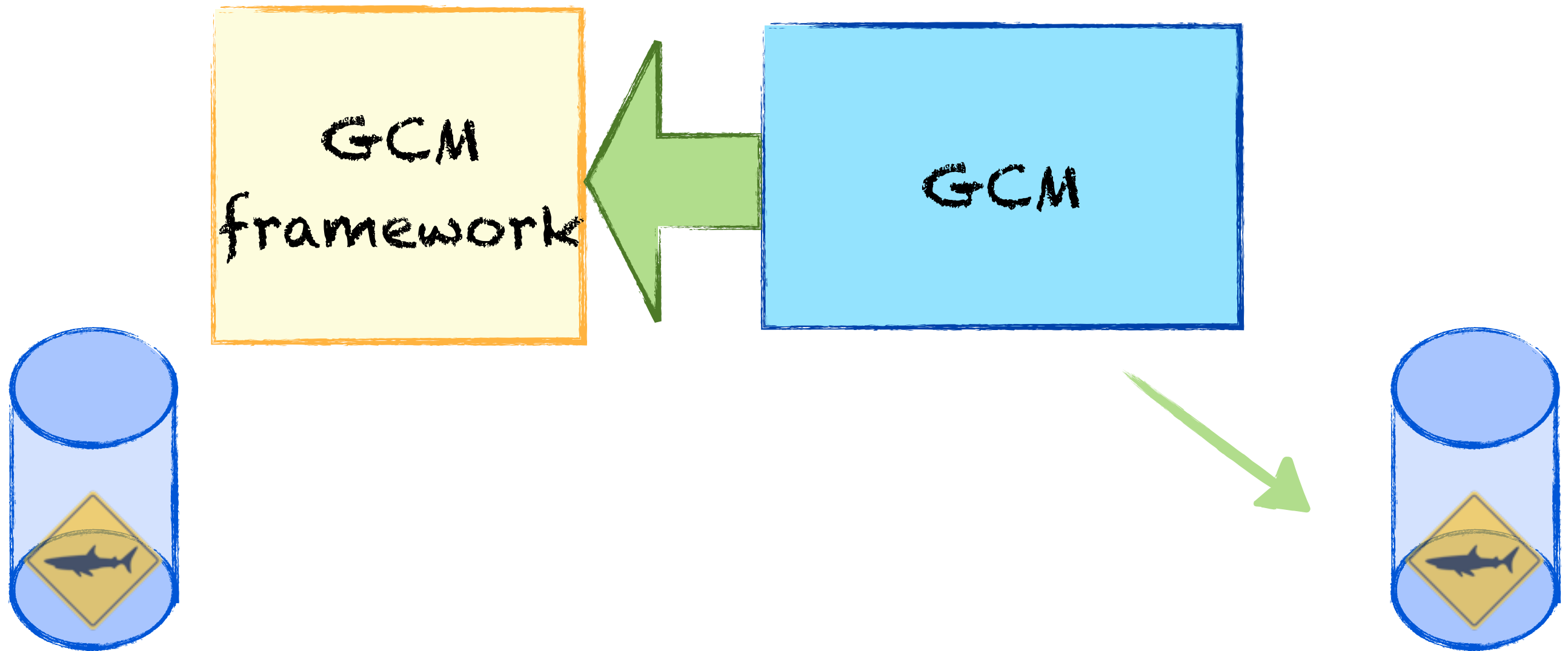
# Deleting ACK'd messages



# Deleting ACK'd messages

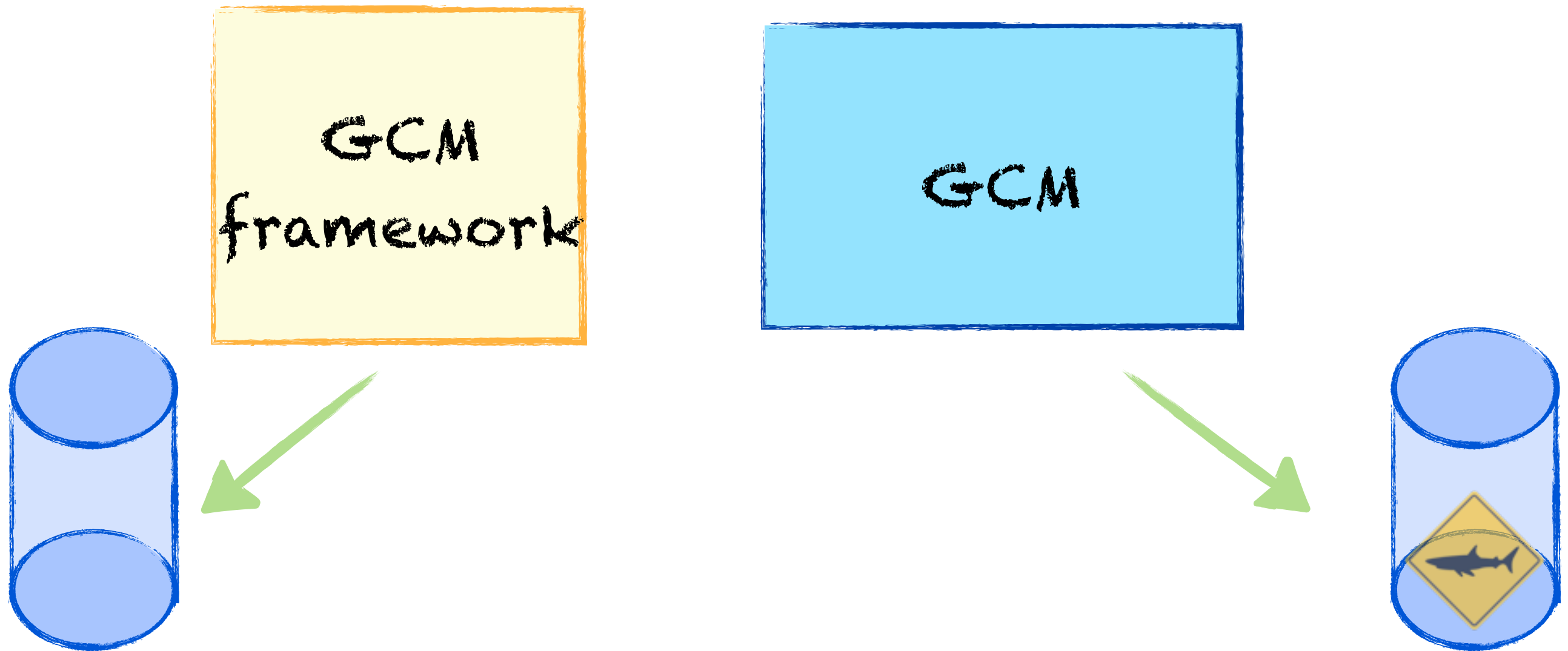


# Deleting ACK'd messages

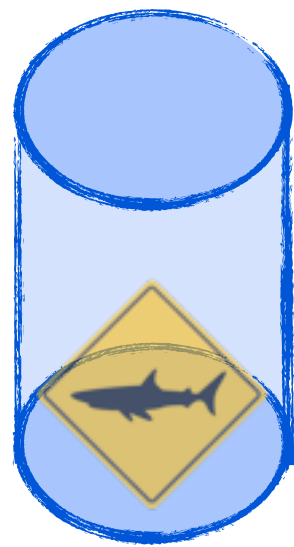




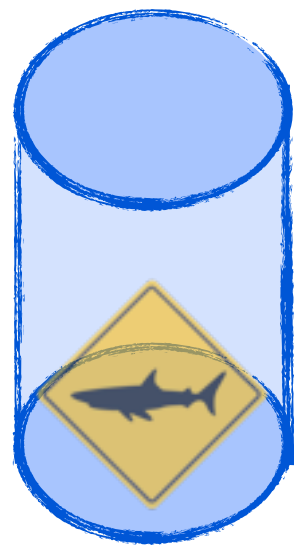
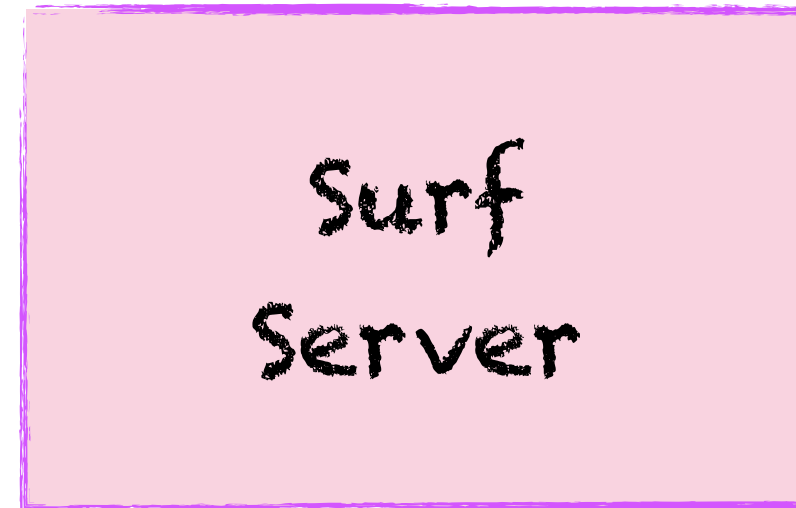
# Deleting ACK'd messages



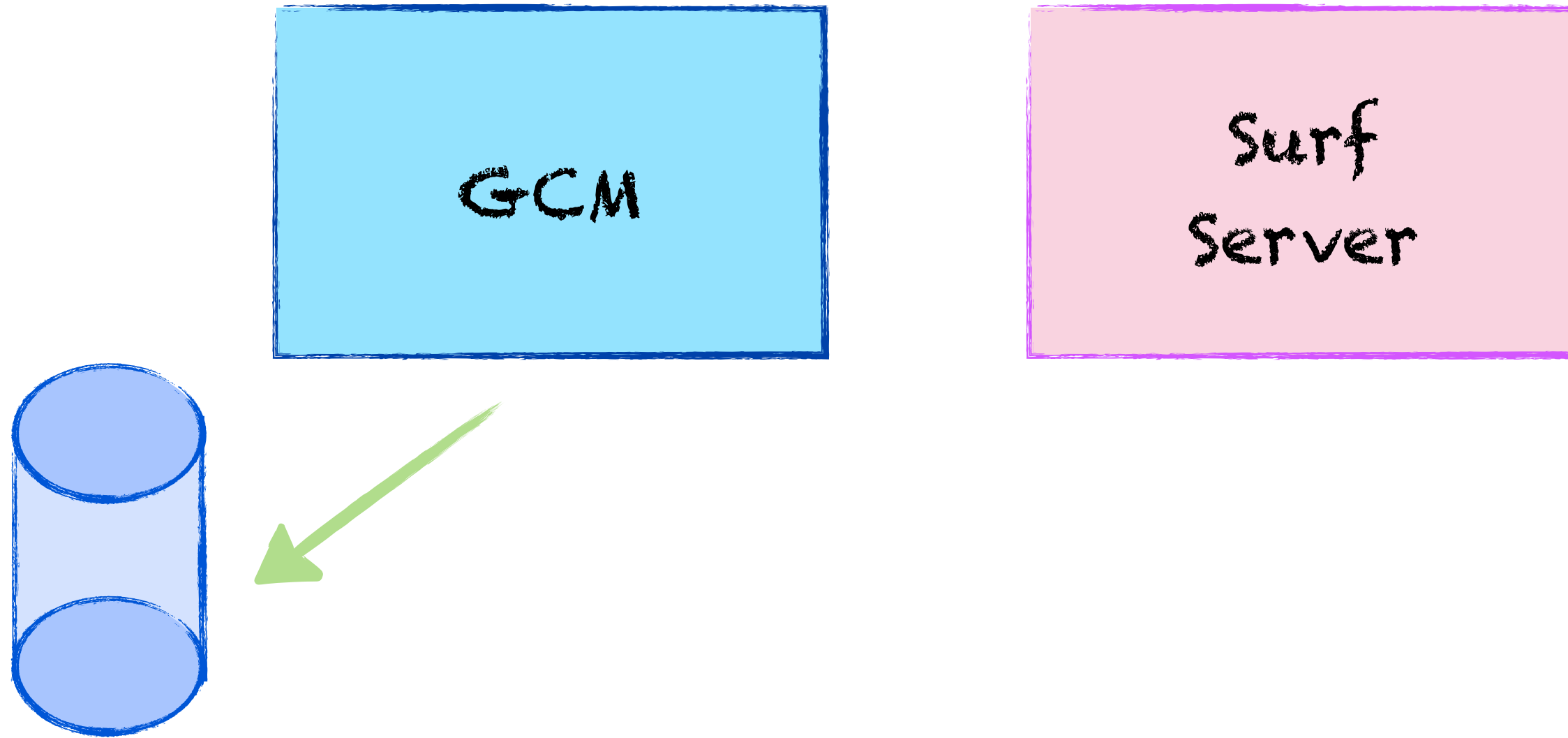
# Deleting ACK'd messages



# Deleting ACK'd messages



# Deleting ACK'd messages



# Google Cloud Messaging **Upstream**

ACKing is required



# Google Cloud Messaging **Upstream**



ACKing is required





# Google Cloud Messaging **Upstream**



ACKing is required



Maximum 100 outstanding msgs



# Google Cloud Messaging **Upstream**



ACKing is required



Maximum 100 outstanding msgs



Maximum 1 day disconnect time



# Battery and Network Optimization

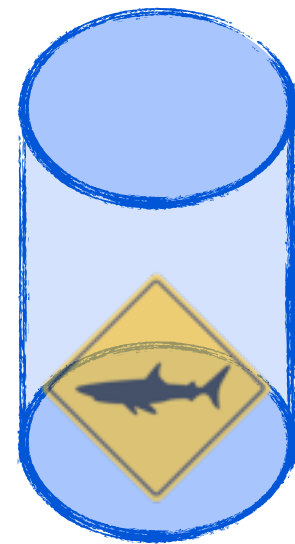
Surf  
App

GCM  
framework

Other  
App



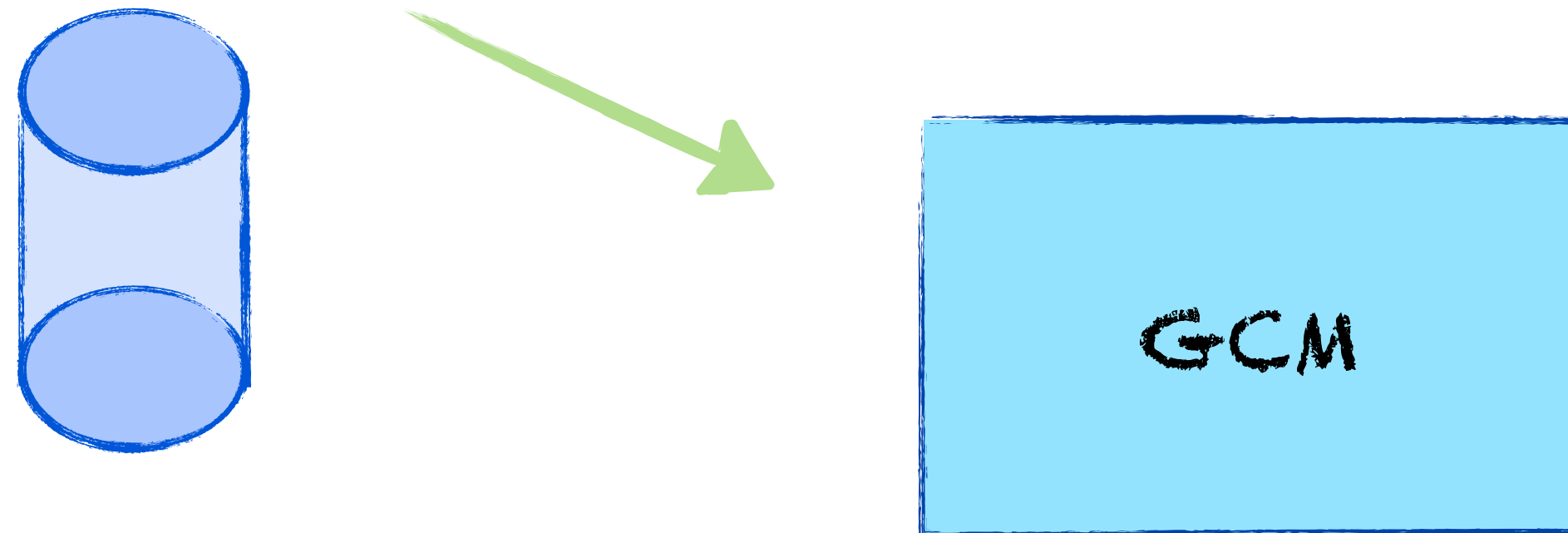
# Battery and Network Optimization



# Battery and Network Optimization

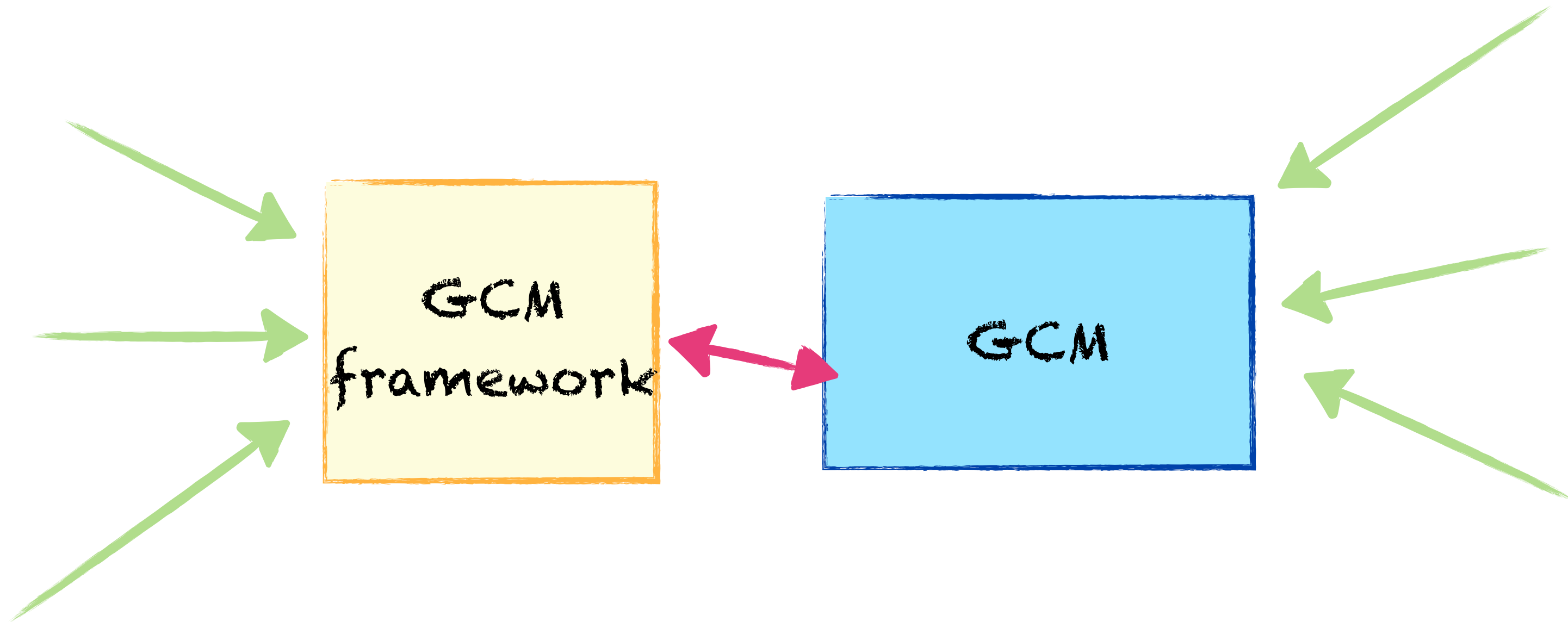


# Battery and Network Optimization





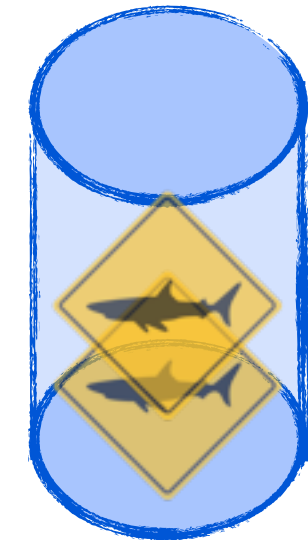
# Battery and Network Optimization



# Battery and Network Optimization

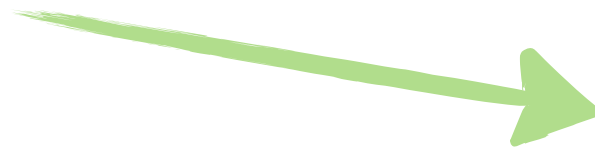
GCM  
framework

GCM

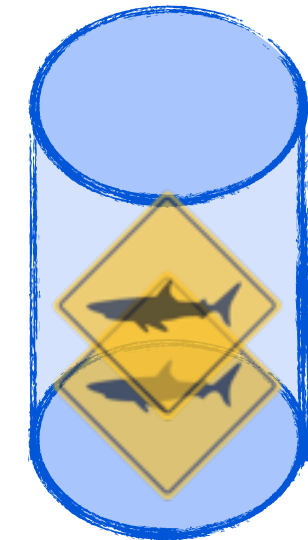
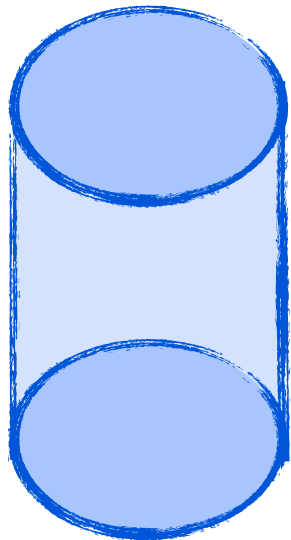


# Battery and Network Optimization

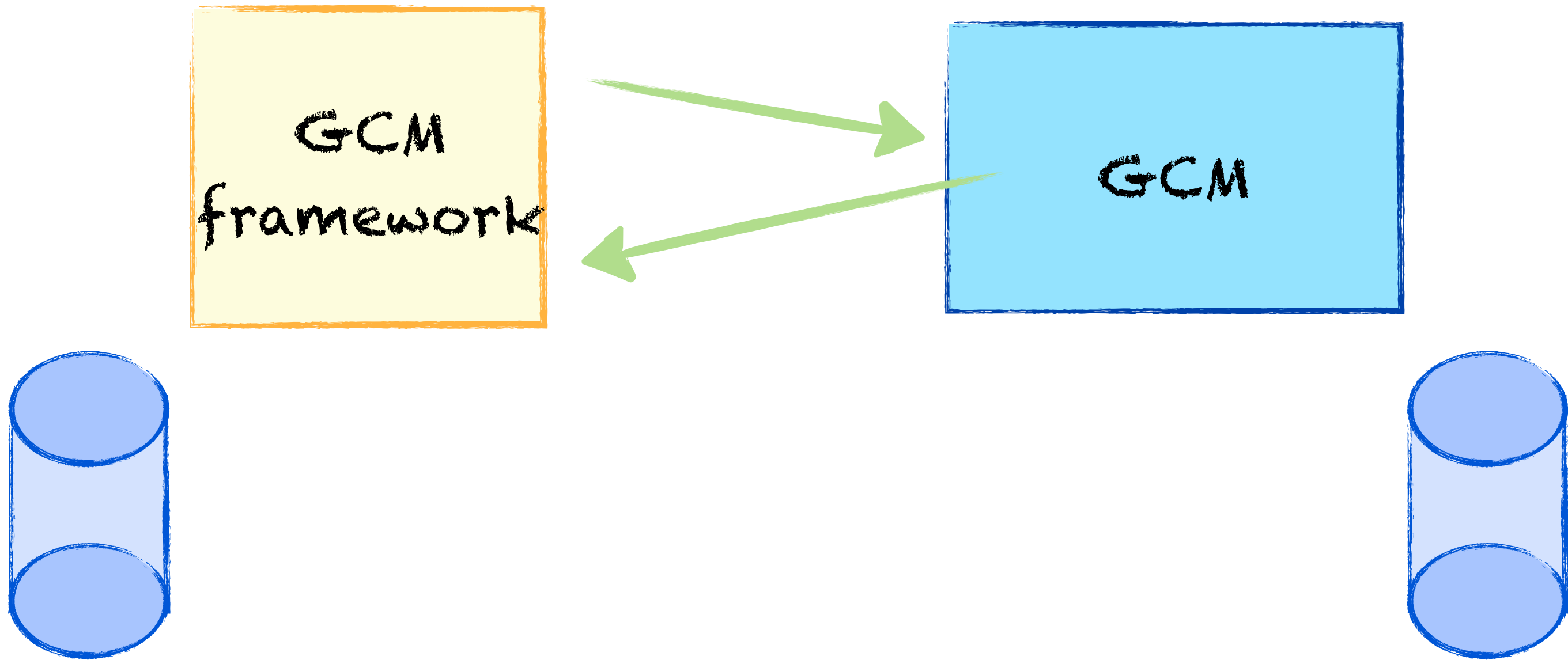
GCM  
framework



GCM



# Battery and Network Optimization



# GCM Upstream

Send messages reliably



# GCM Upstream



Send messages reliably





# GCM Upstream



Send messages reliably



Send messages with low latency



# GCM Upstream



Send messages reliably



Send messages with low latency



Time To Live API







Google Cloud Messaging  
meets

Google Play Services



# GCM library meets *Play Services*

Old Library Compatible



# GCM library meets Play Services



Old Library Compatible



# GCM library meets **Play Services**



Old Library Compatible



New APIs from Froyo+





# GCM library meets *Play Services*



Old Library Compatible



New APIs from Froyo+



Upstream only on Play Services



```
GoogleCloudMessaging gcm =  
    GoogleCloudMessaging.getInstance(context);  
  
String registrationId = gcm.register(sender1, sender2);
```



```
<uses-permission  
    android:name="com.google.android.c2dm.permission.RECEIVE" />  
  
<receiver android:name=".MyReceiver" android:exported="true"  
    android:permission="com.google.android.c2dm.permission.SEND" >  
    <intent-filter>  
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />  
        <category android:name="YOUR_PACKAGE_NAME" />  
    </intent-filter>  
</receiver>
```

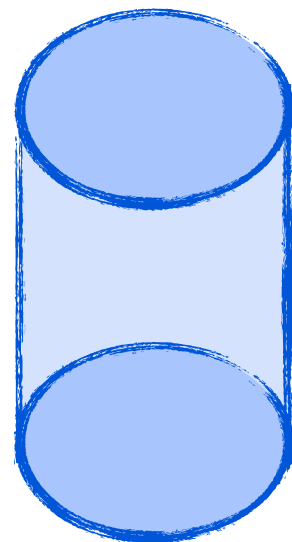


# Registrations are cached on **Play Services**

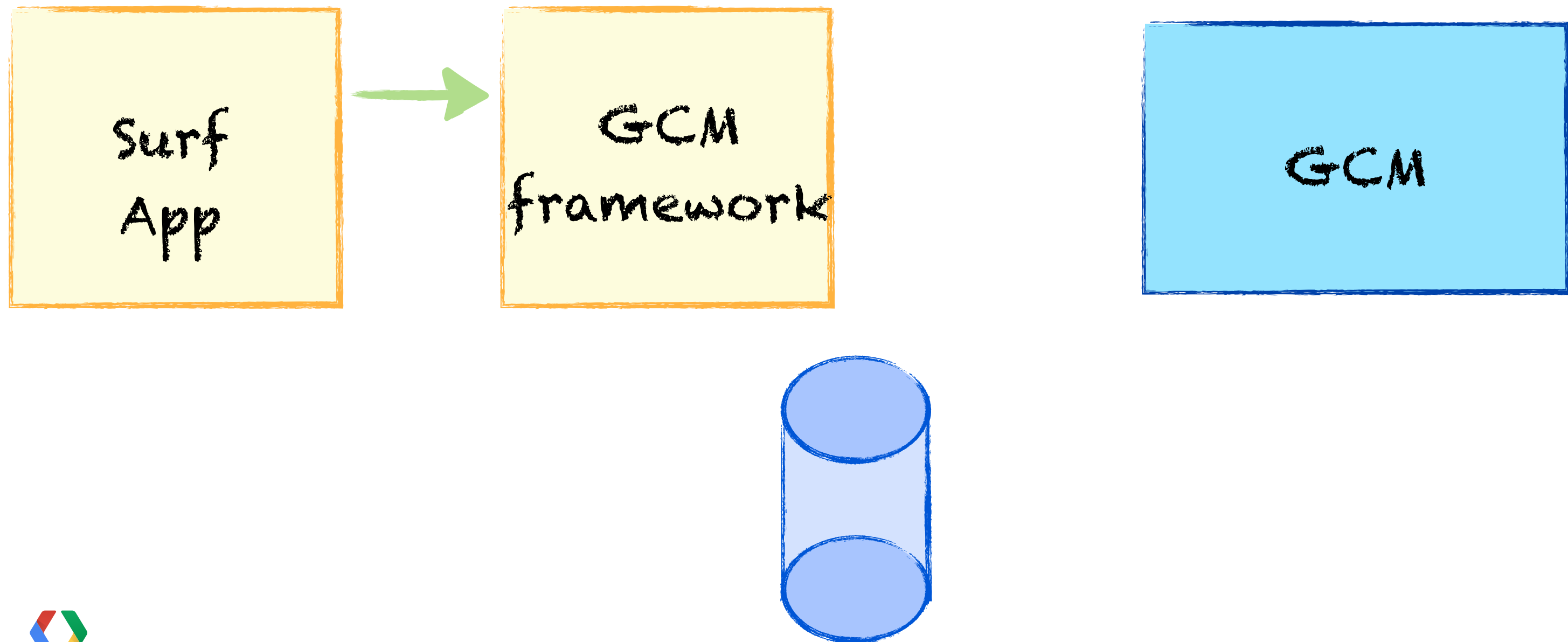
Surf  
App

GCM  
framework

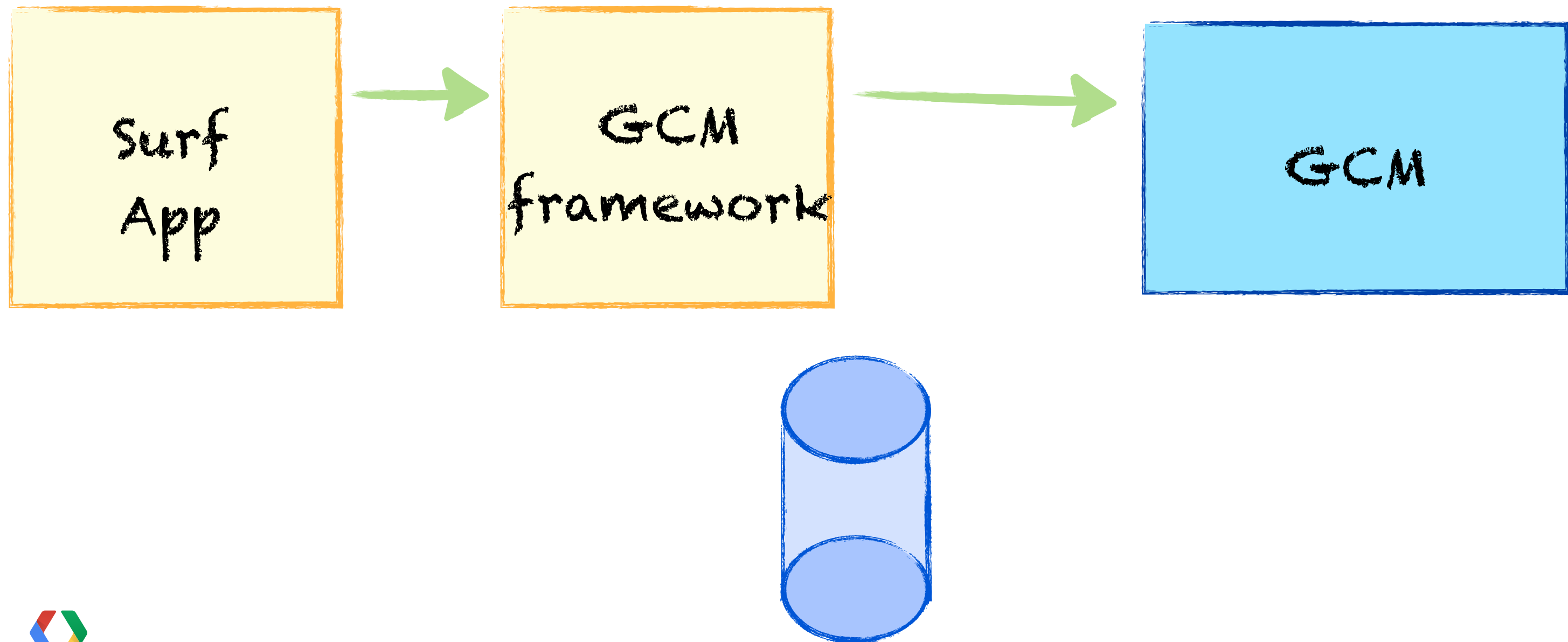
GCM



# Registrations are cached on **Play Services**

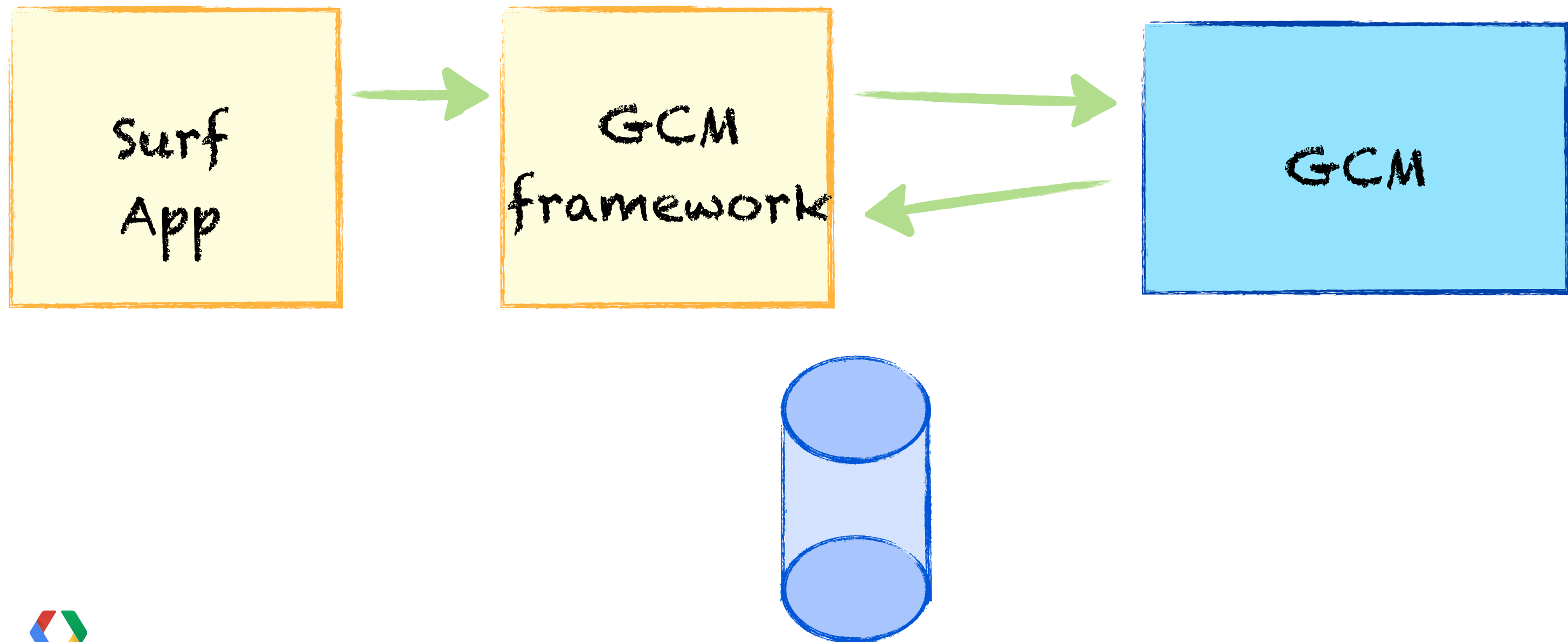


# Registrations are cached on **Play Services**

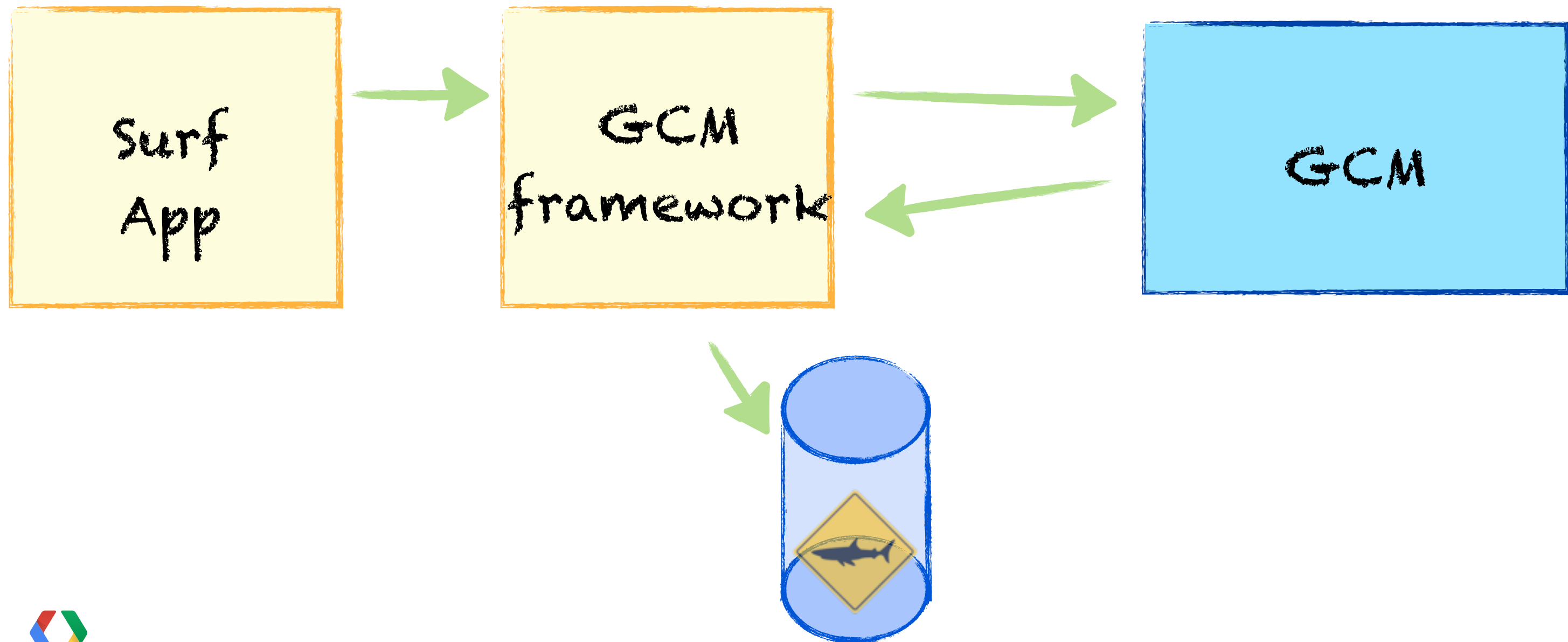




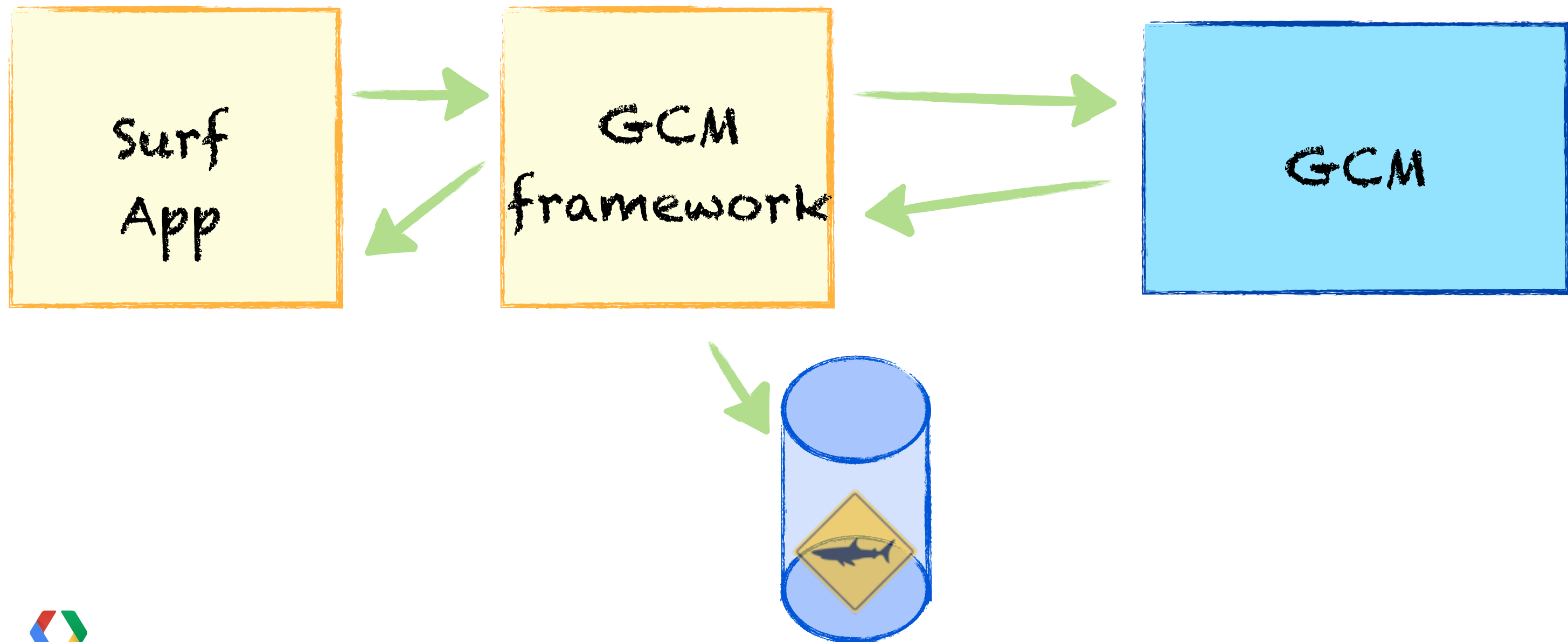
# Registrations are cached on **Play Services**



# Registrations are cached on **Play Services**



# Registrations are cached on **Play Services**

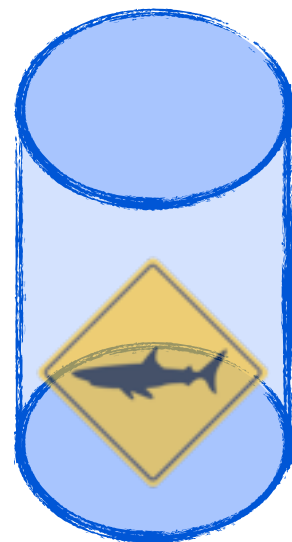


# Registrations are cached on **Play Services**

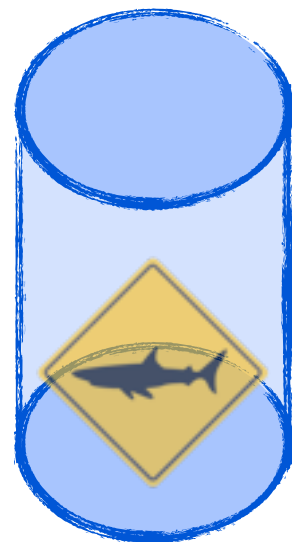
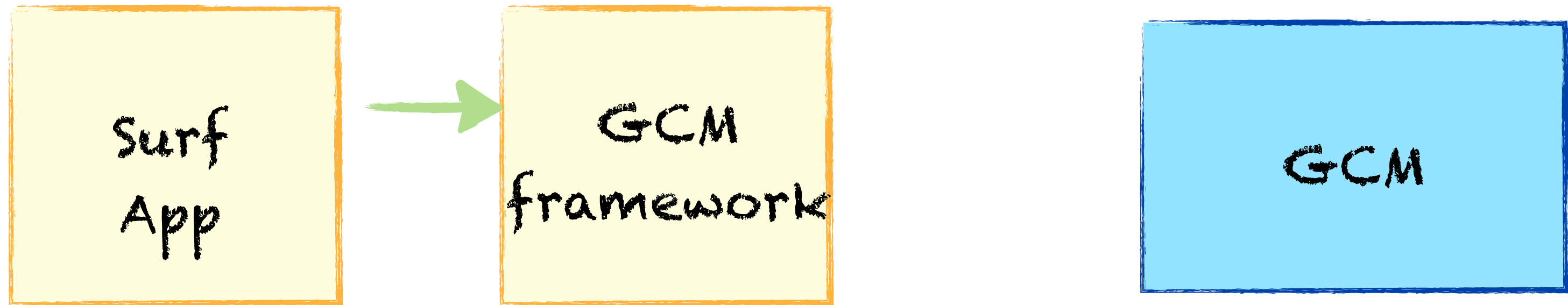
Surf  
App

GCM  
framework

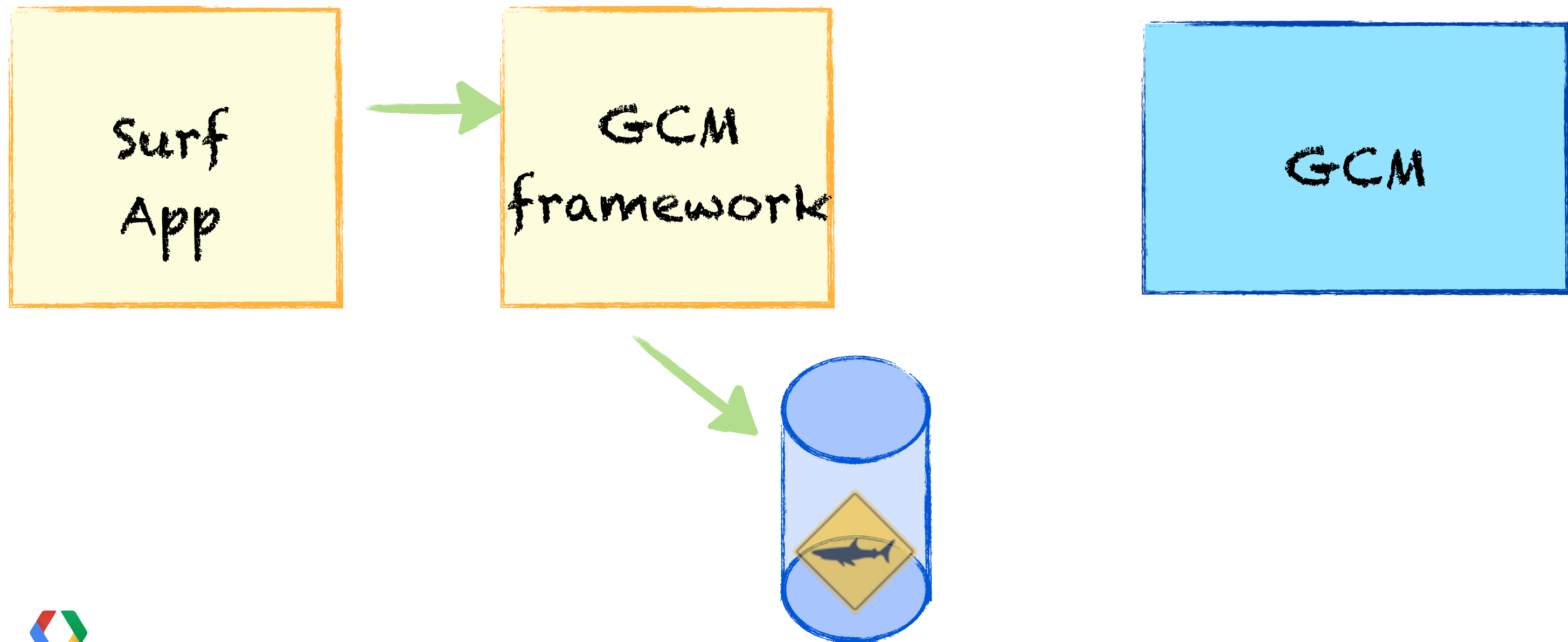
GCM



# Registrations are cached on **Play Services**

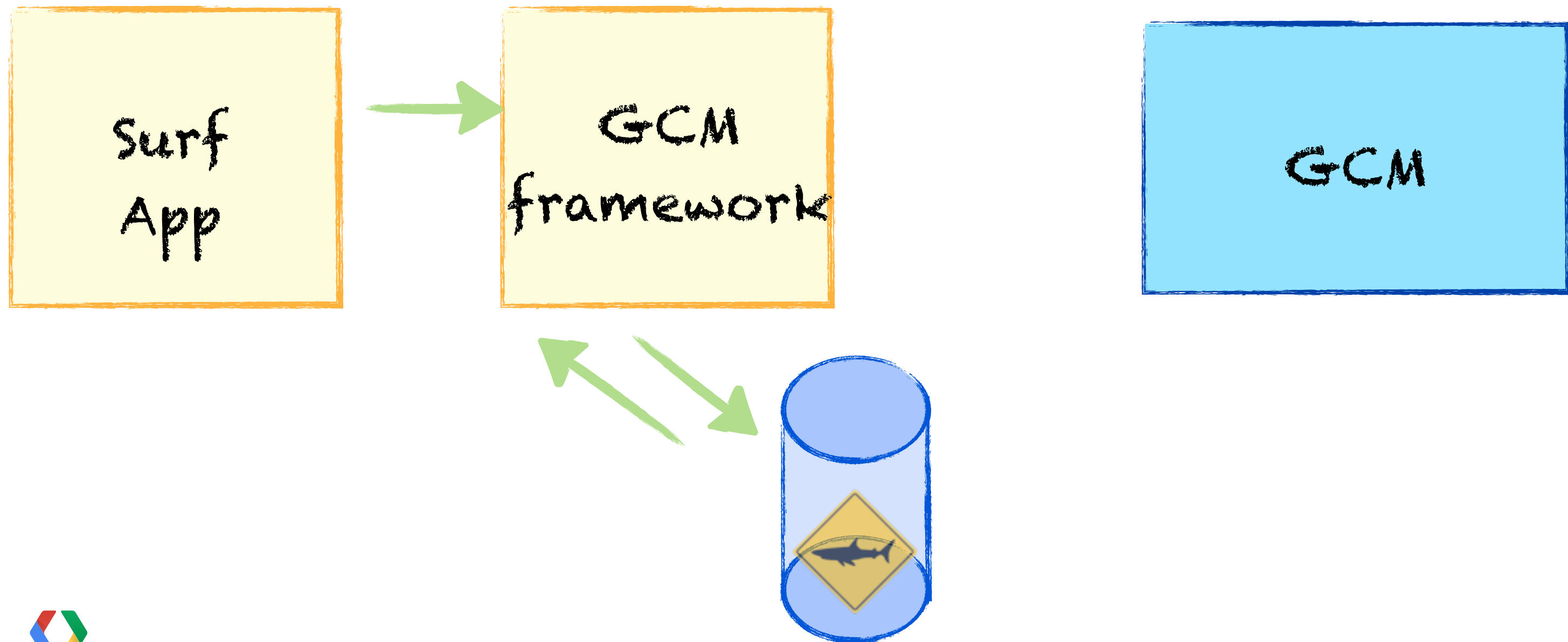


# Registrations are cached on **Play Services**

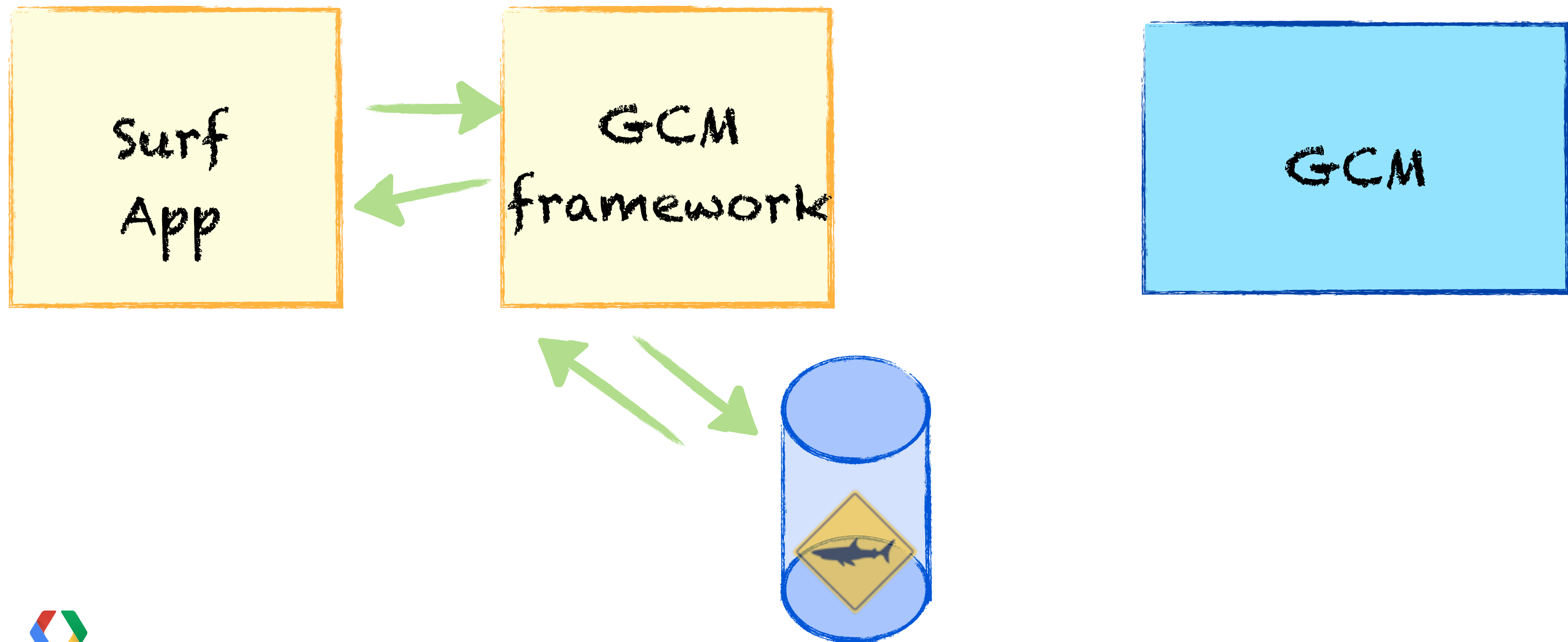




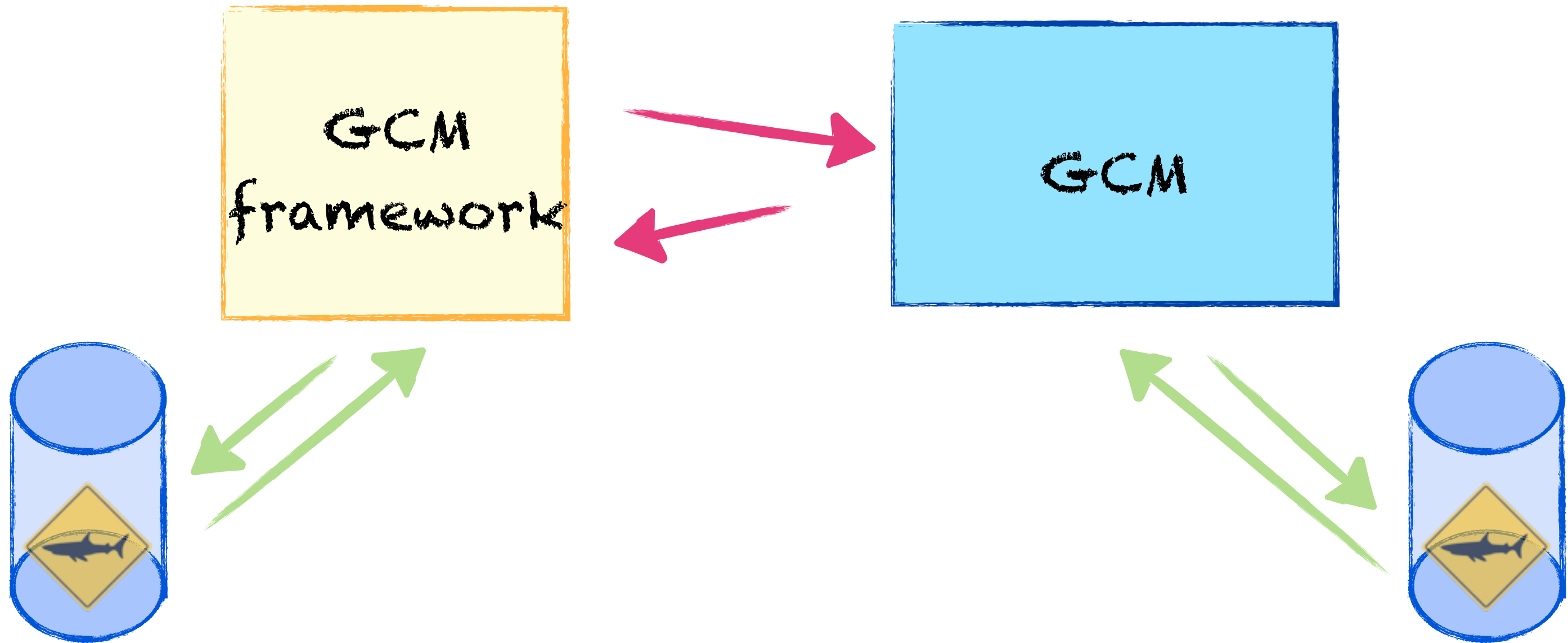
# Registrations are cached on **Play Services**



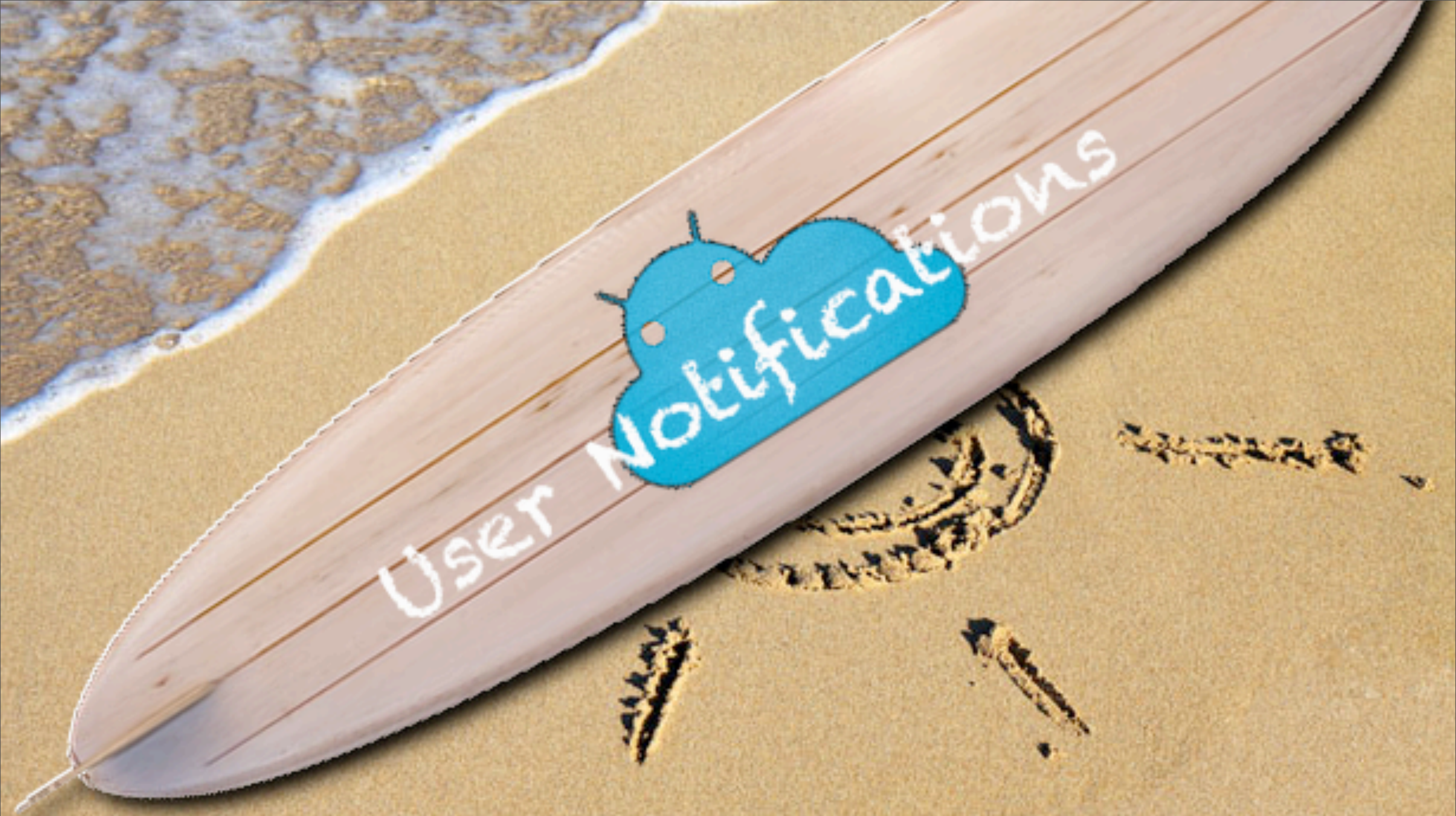
# Registrations are cached on **Play Services**



# Registrations are periodically *sync'd*









# Google Cloud Messaging User Notifications

Surf  
Server

GCM

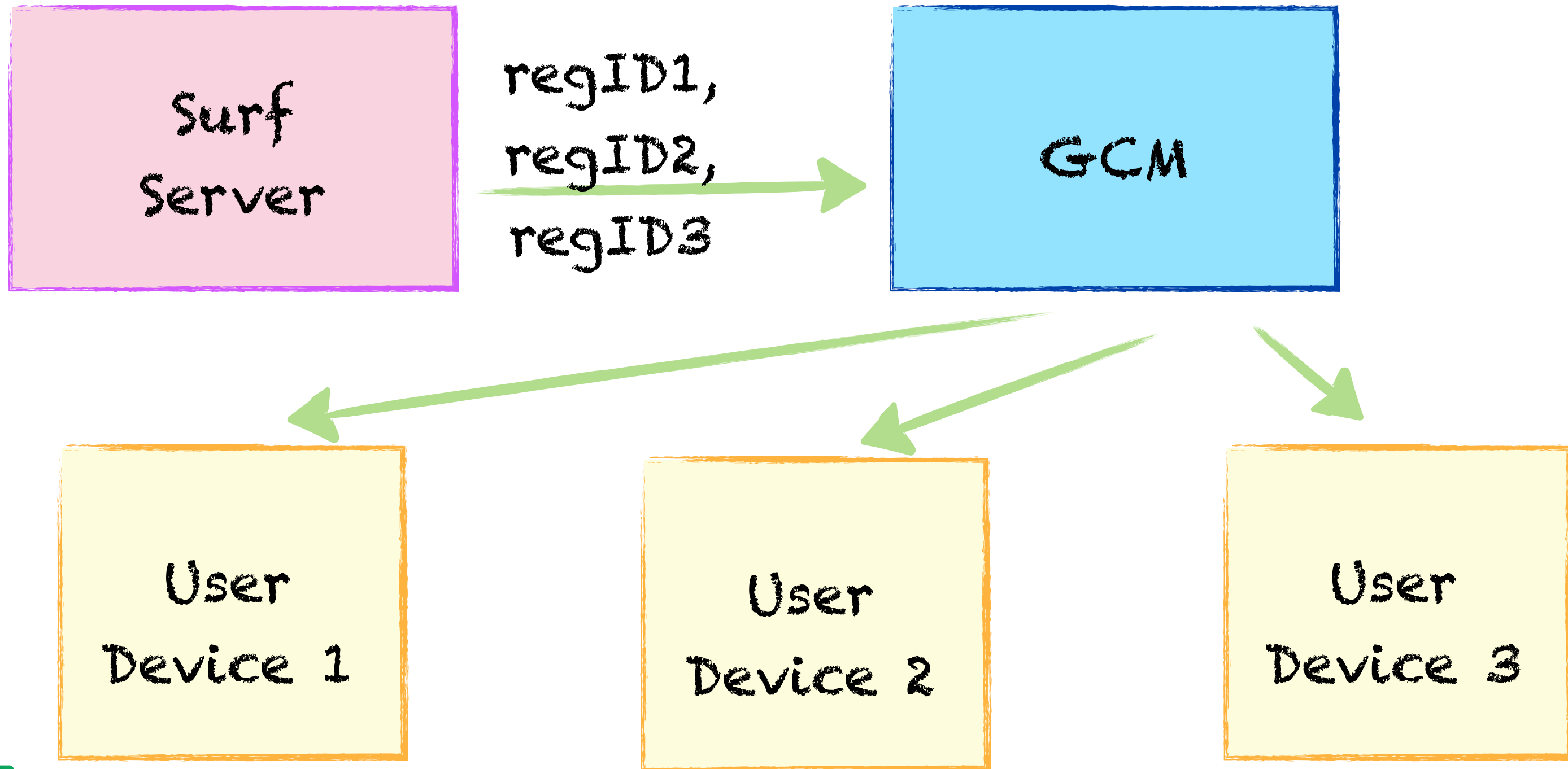
User  
Device 1

User  
Device 2

User  
Device 3



# Google Cloud Messaging User Notifications





# Google Cloud Messaging **User Notifications**

Surf  
Server

GCM

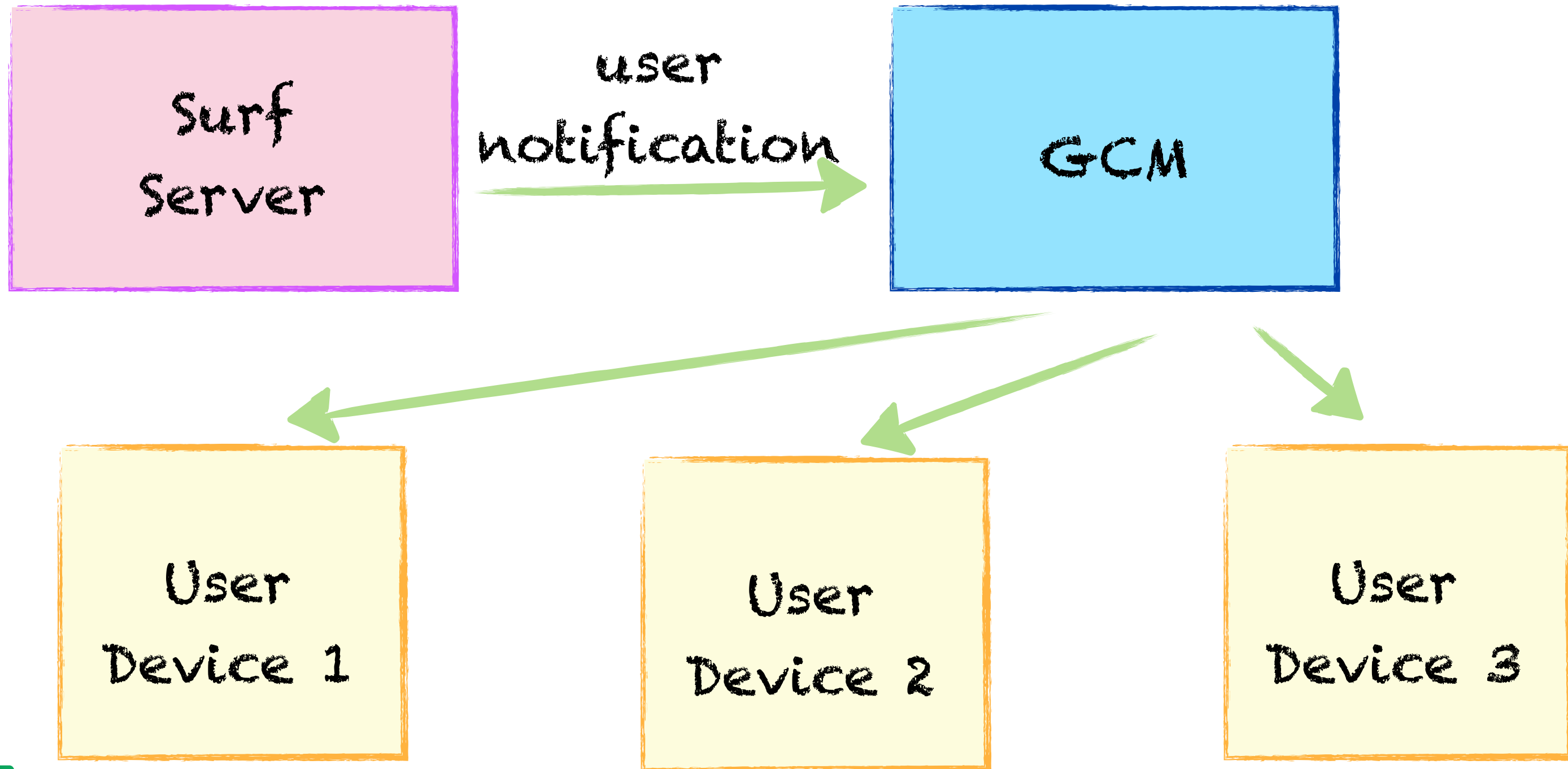
User  
Device 1

User  
Device 2

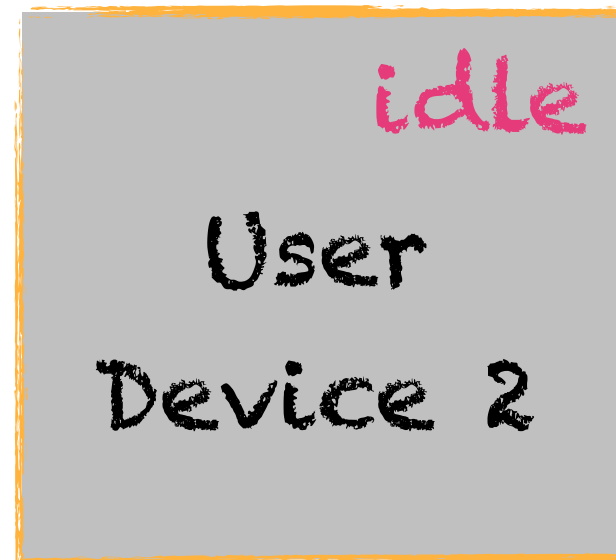
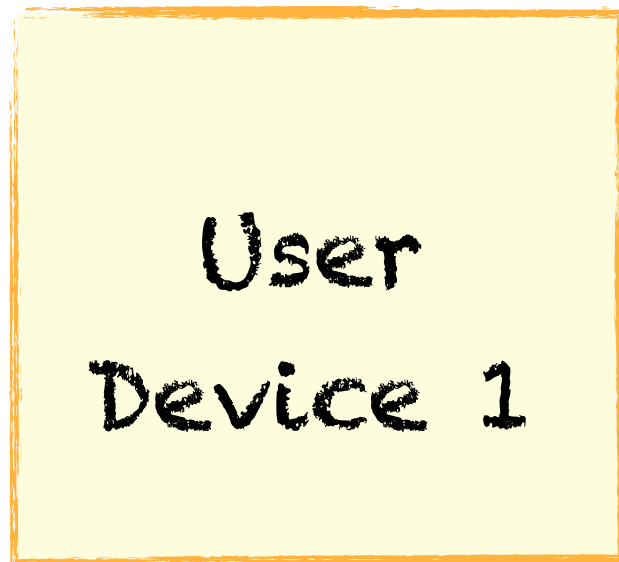
User  
Device 3



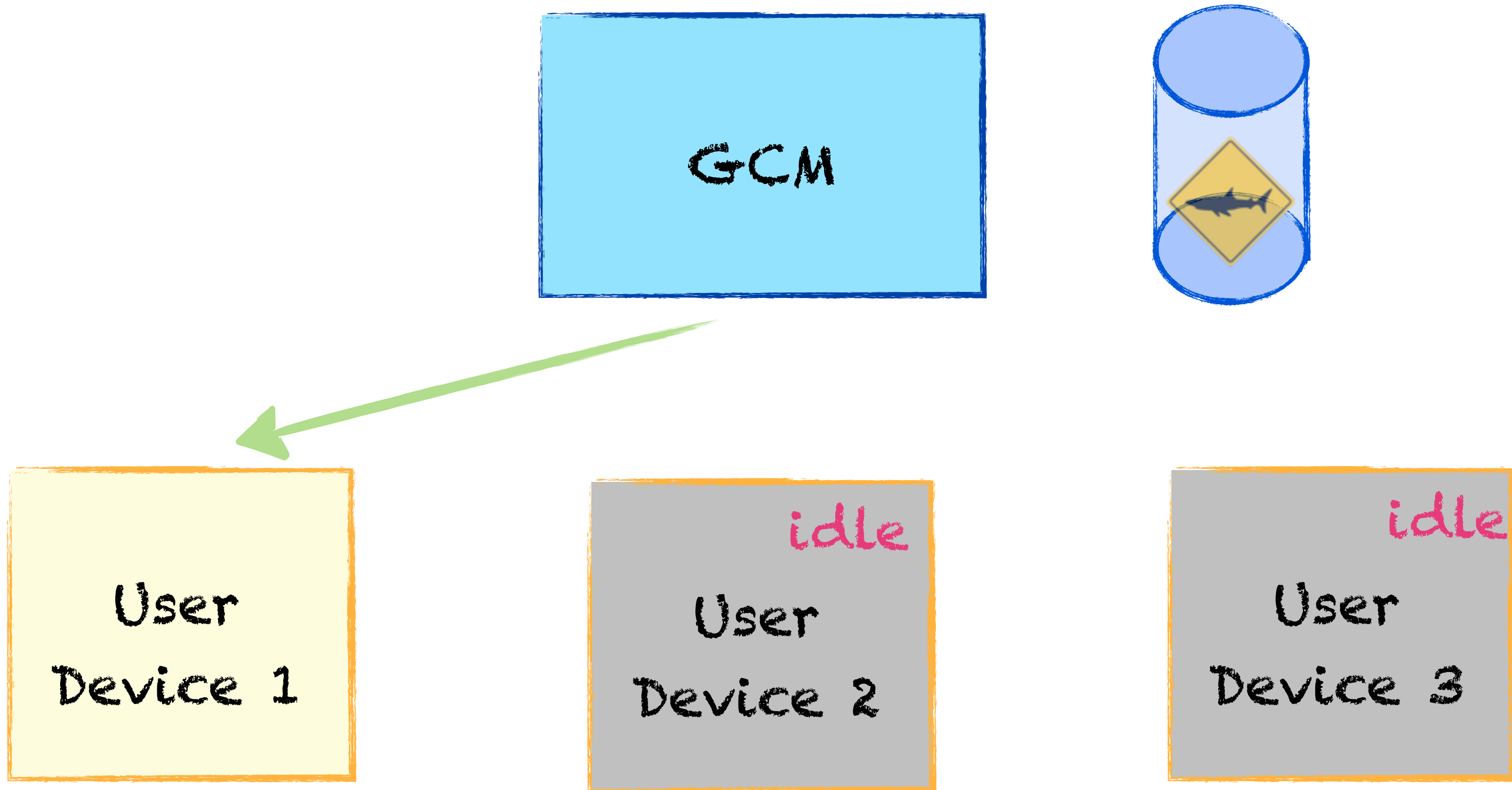
# Google Cloud Messaging User Notifications



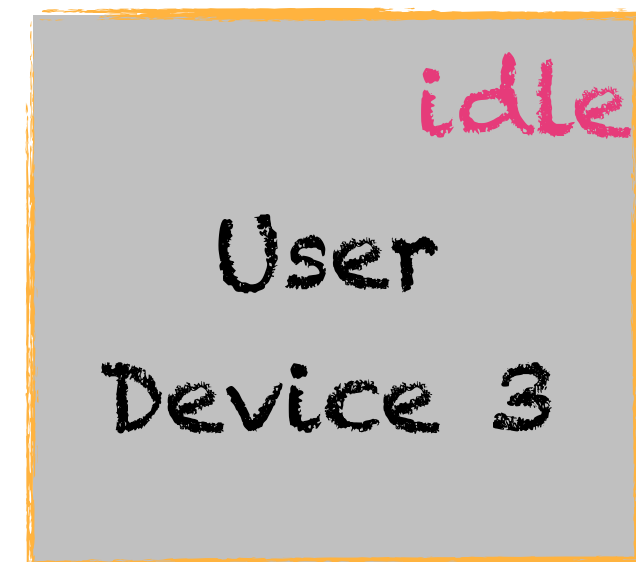
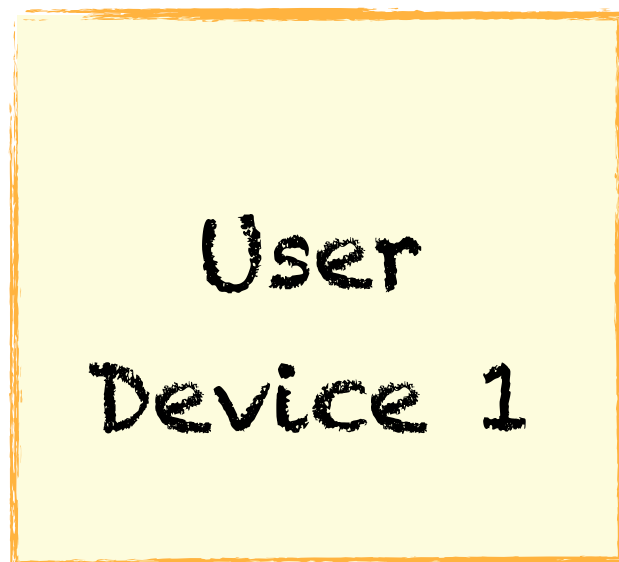
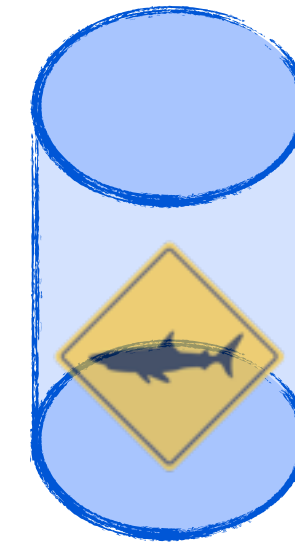
# Google Cloud Messaging *User Notifications*



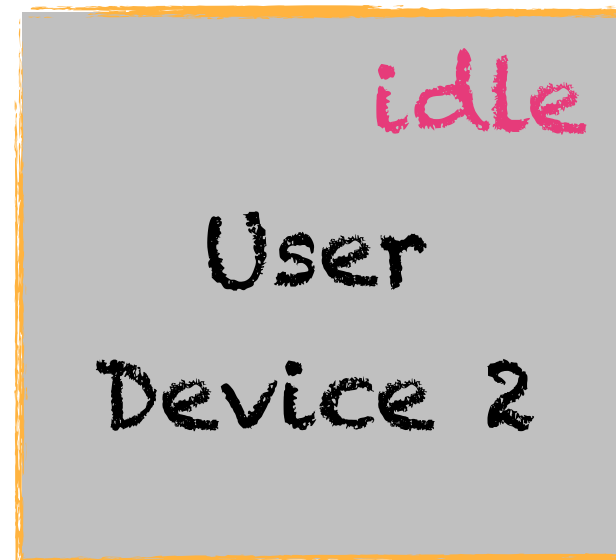
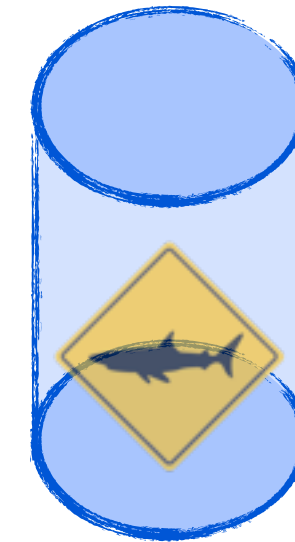
# Google Cloud Messaging User Notifications



# Google Cloud Messaging *User Notifications*

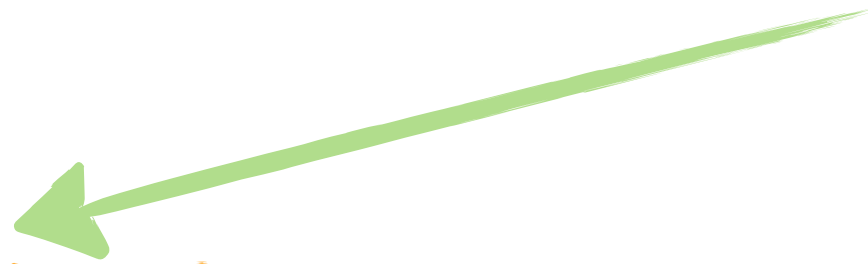
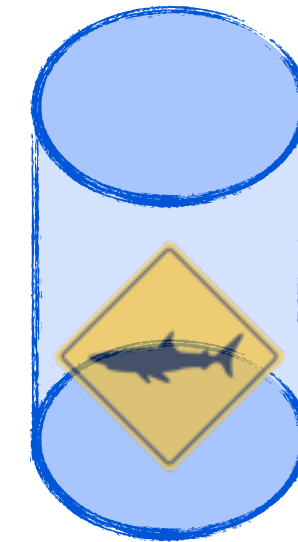


# Google Cloud Messaging User Notifications

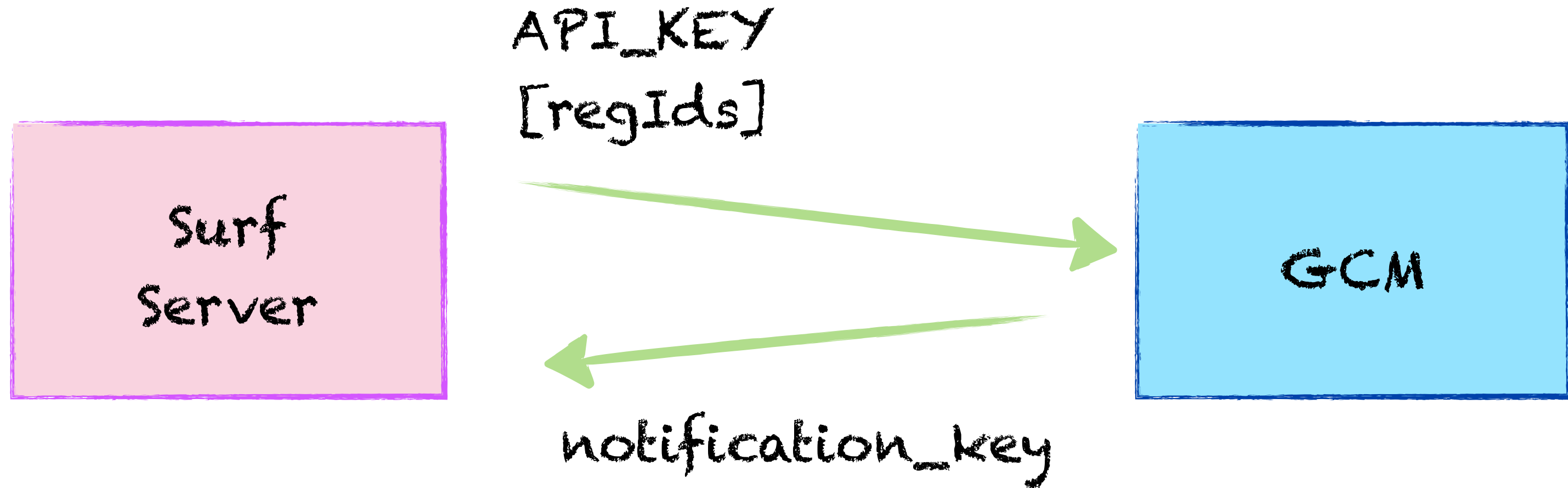




# Google Cloud Messaging *User Notifications*



# Google Cloud Messaging User Notifications



**<https://android.googleapis.com/gcm/notifications>**

**content-type: "application/json"**

**Header : "project\_id": <projectID>**

**Header: "Authorization", "key=API\_KEY"**



## Request:

```
{  
  "operation": ["create", "add", "remove"],  
  "notification_key_name": "user1_hash",  
  "registration_ids": ["XYZ", "ABC"]  
}
```

## Response:

```
{  
  "notification_key": "a8YTG63hHH",  
}
```



```
{  
  "to": "NOTIFICATION_KEY",  
  "data": {"hello": "world", }  
  "collapse_key": "collapse"  
  "time_to_live": "600",  
  "message_id": "m-1366082849205"  
  "delay_while_idle": true/false  
}
```



# Google Cloud Messaging **User Notifications**

Surf  
Server

GCM

User  
Device 1

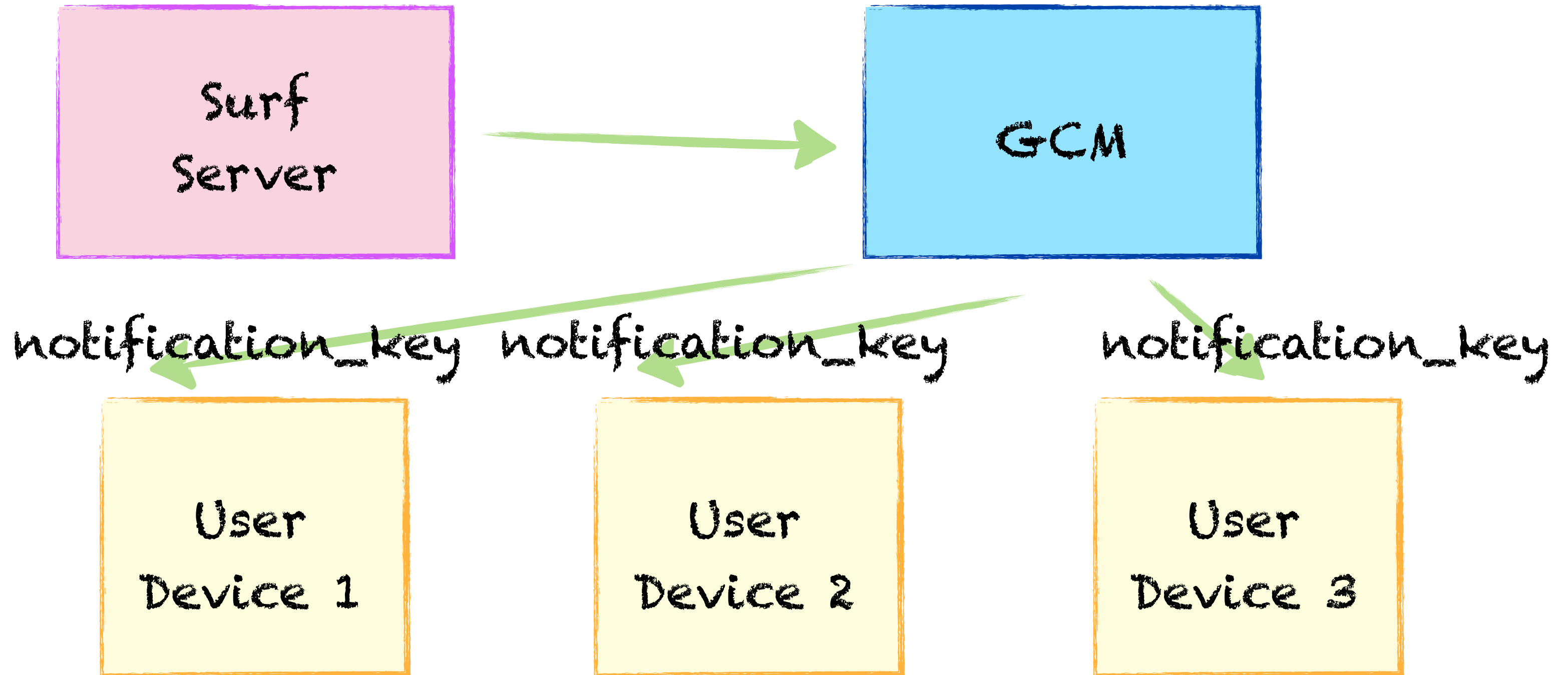
User  
Device 2

User  
Device 3





# Google Cloud Messaging User Notifications



# Google Cloud Messaging **User Notifications**

GCM

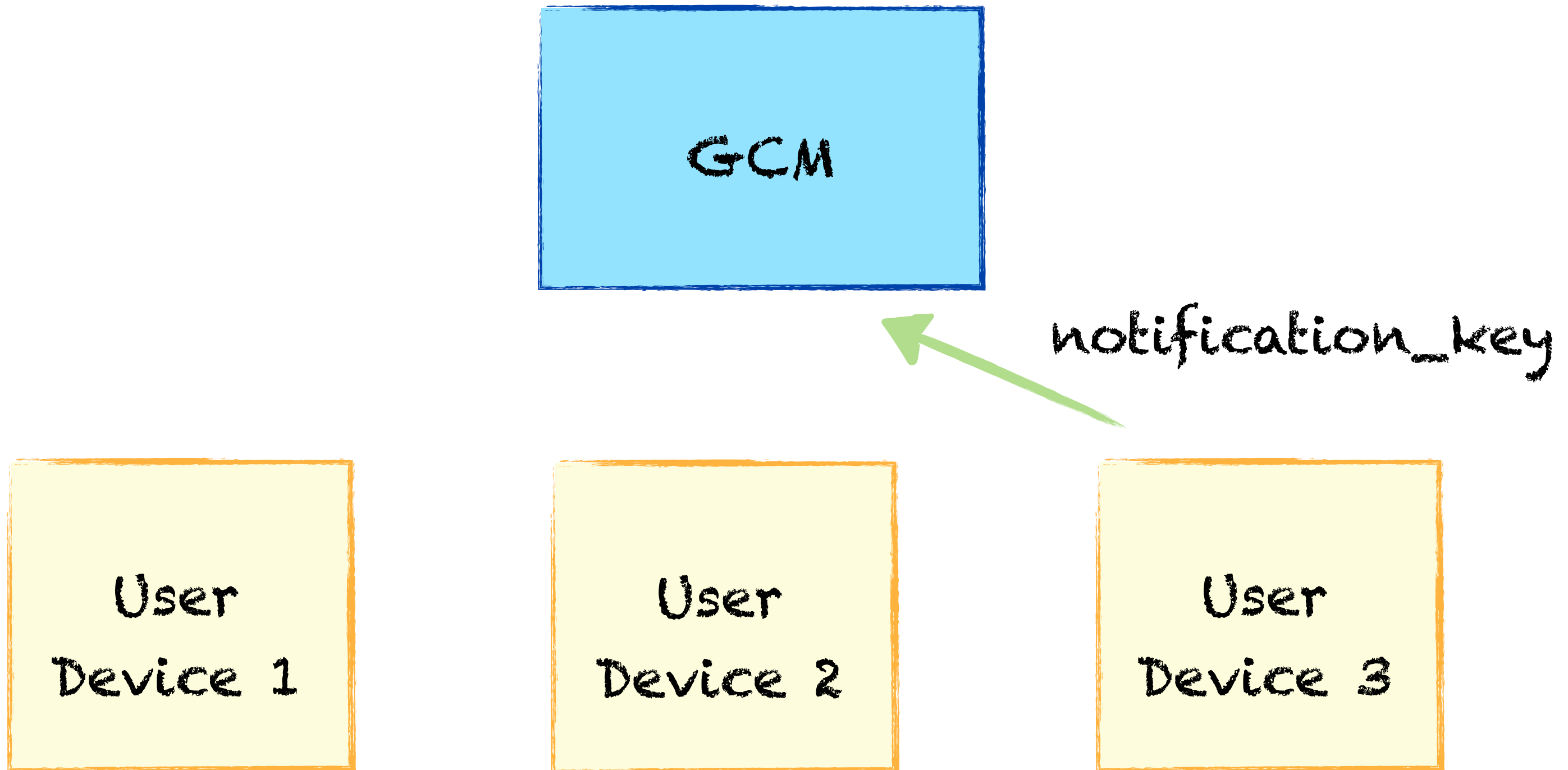
User  
Device 1

User  
Device 2

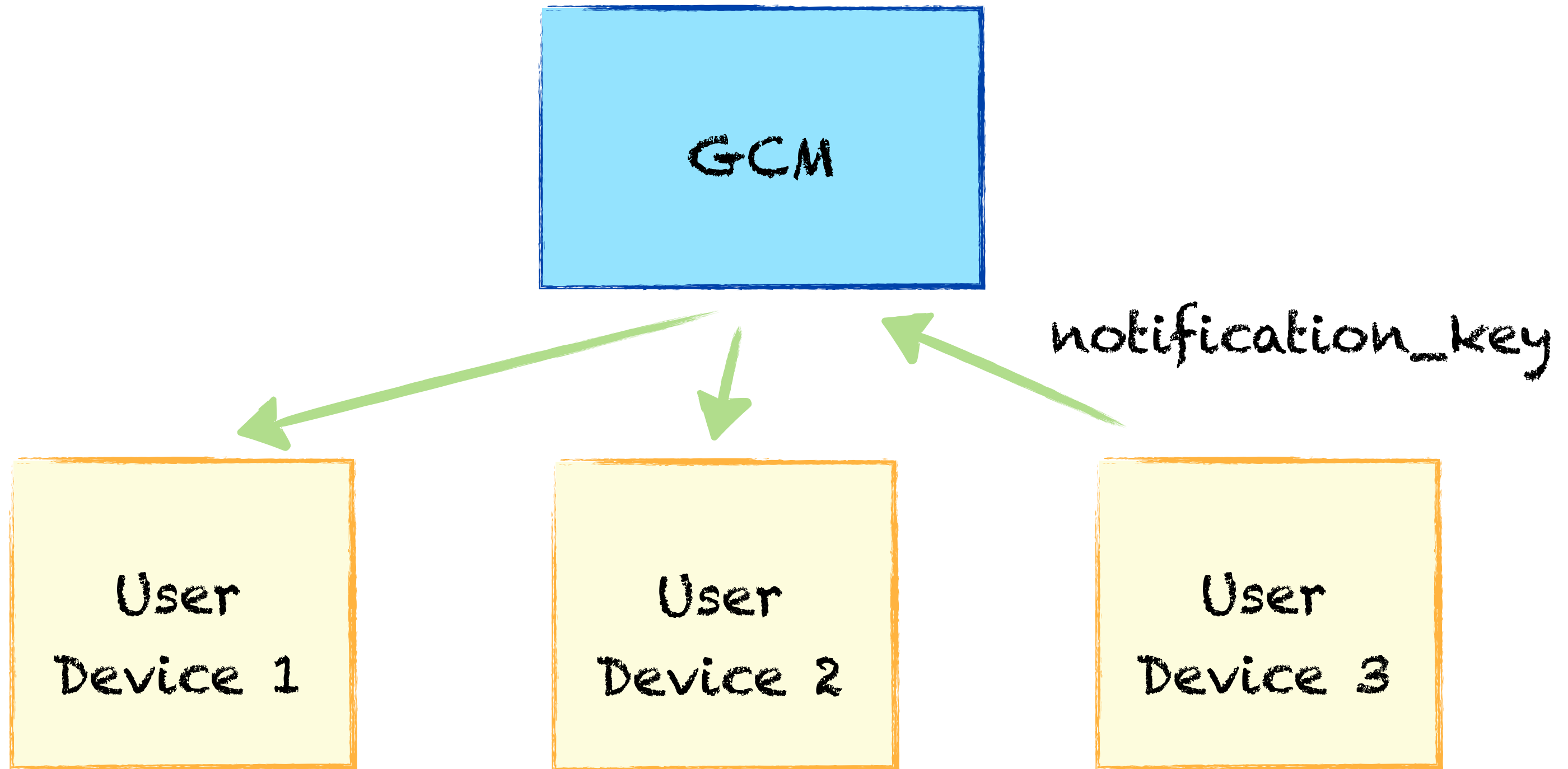
User  
Device 3



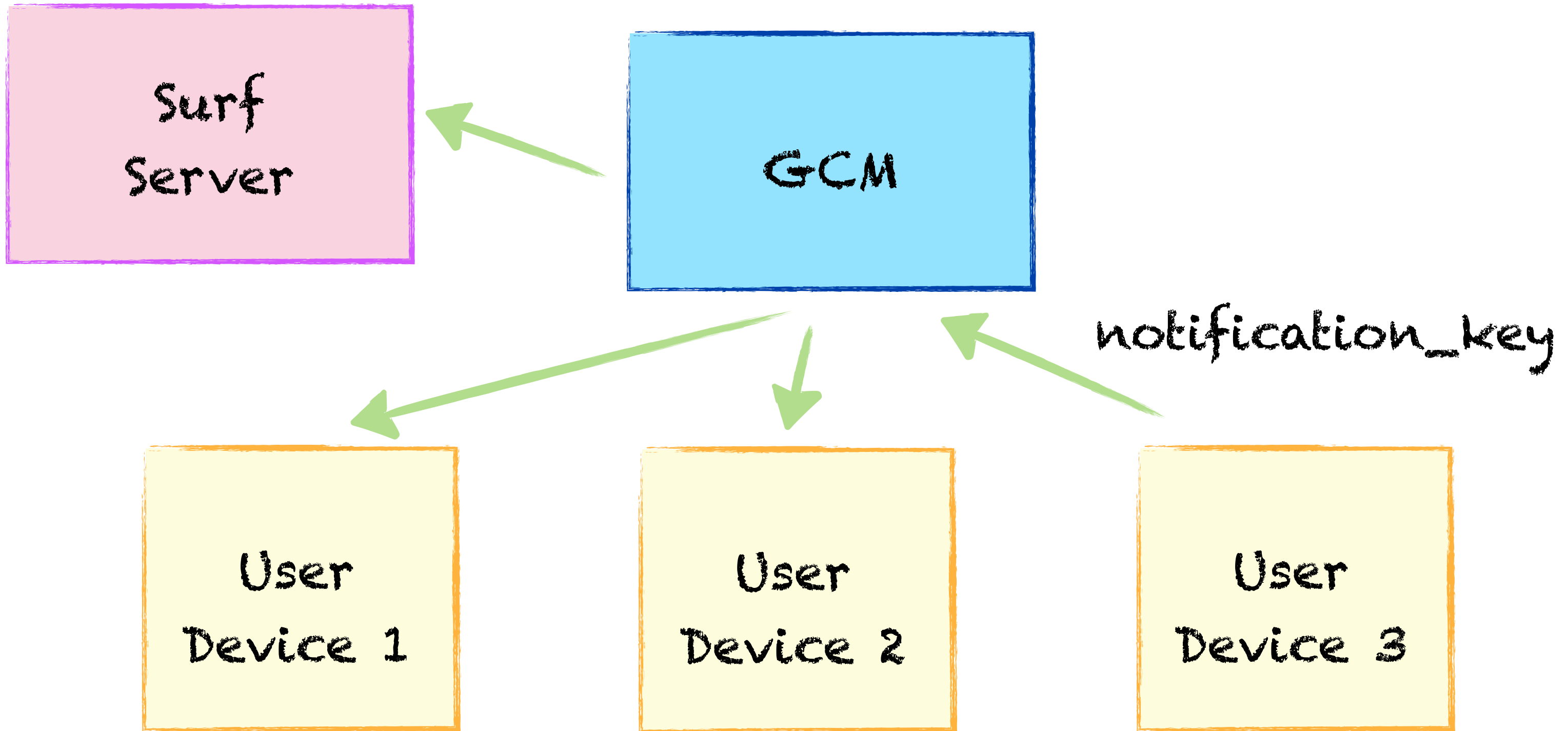
# Google Cloud Messaging **User Notifications**



# Google Cloud Messaging **User Notifications**



# Google Cloud Messaging **User Notifications**



```
GoogleCloudMessaging gcm =  
    GoogleCloudMessaging.get(context);
```

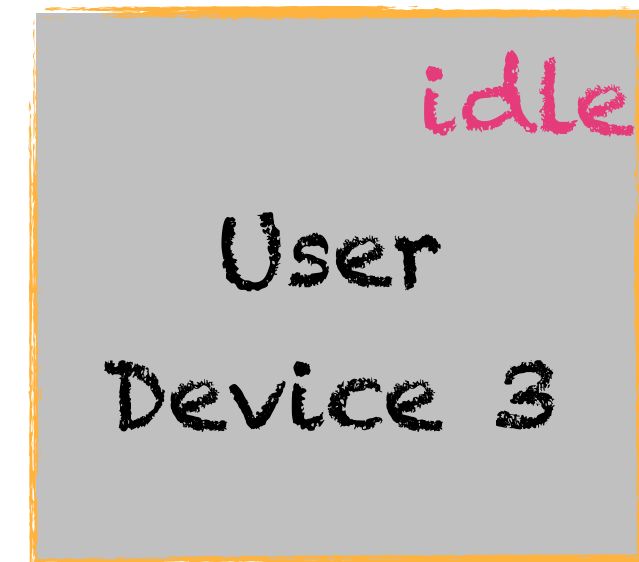
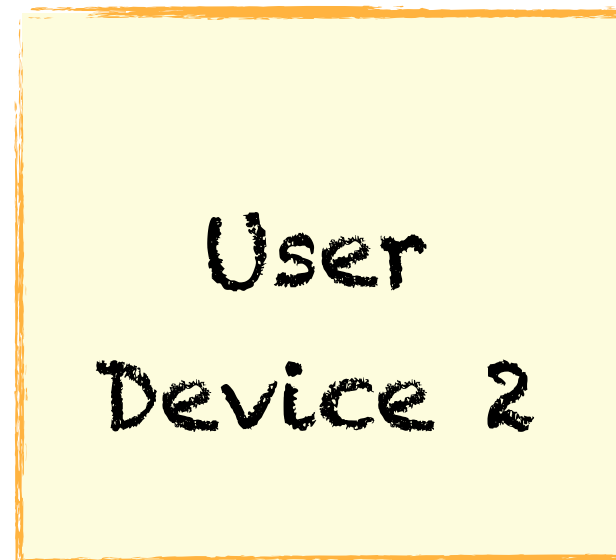
**String to = NOTIFICATION\_KEY**

```
gcm.send(to, msgId, data);
```

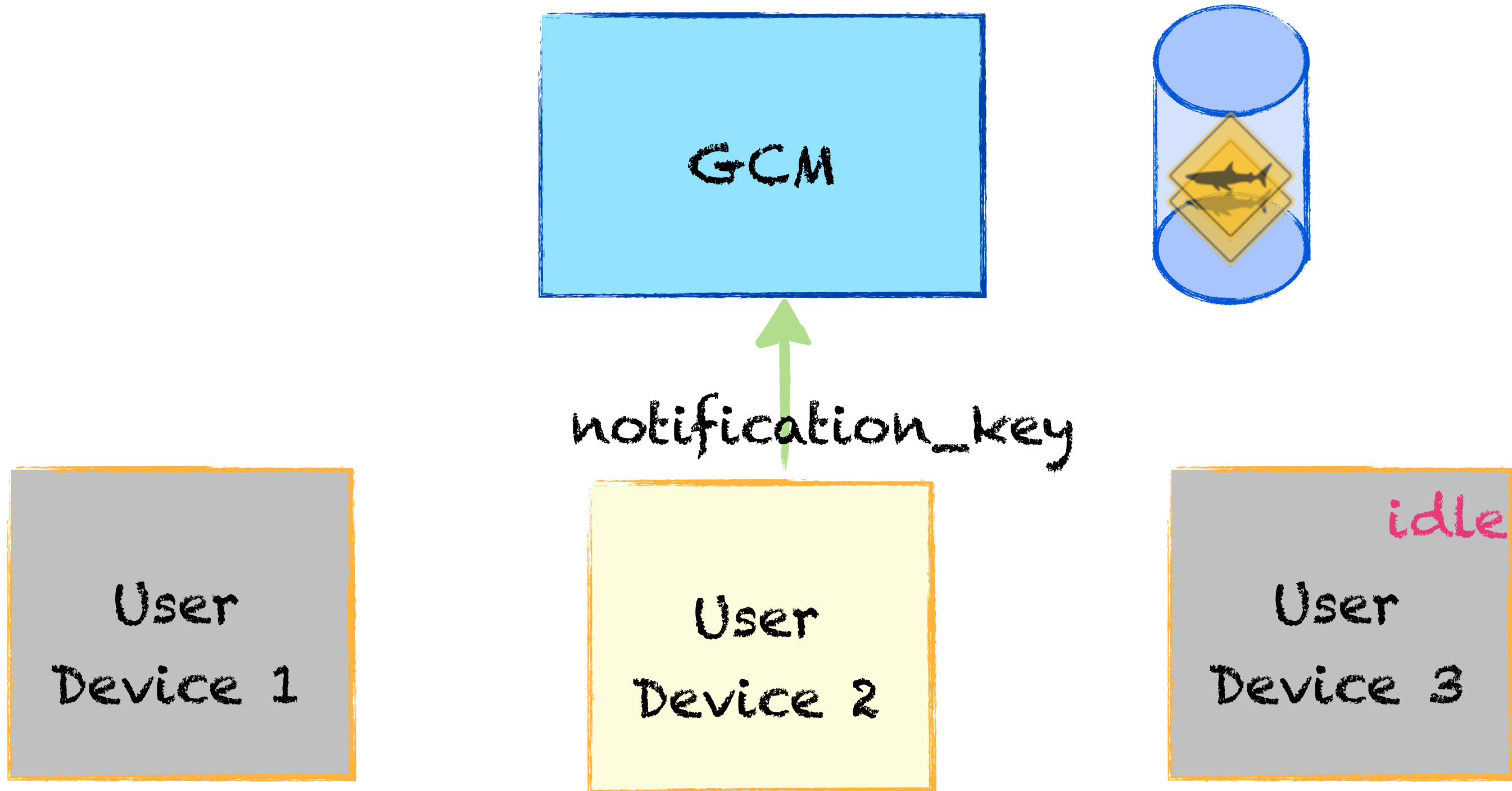




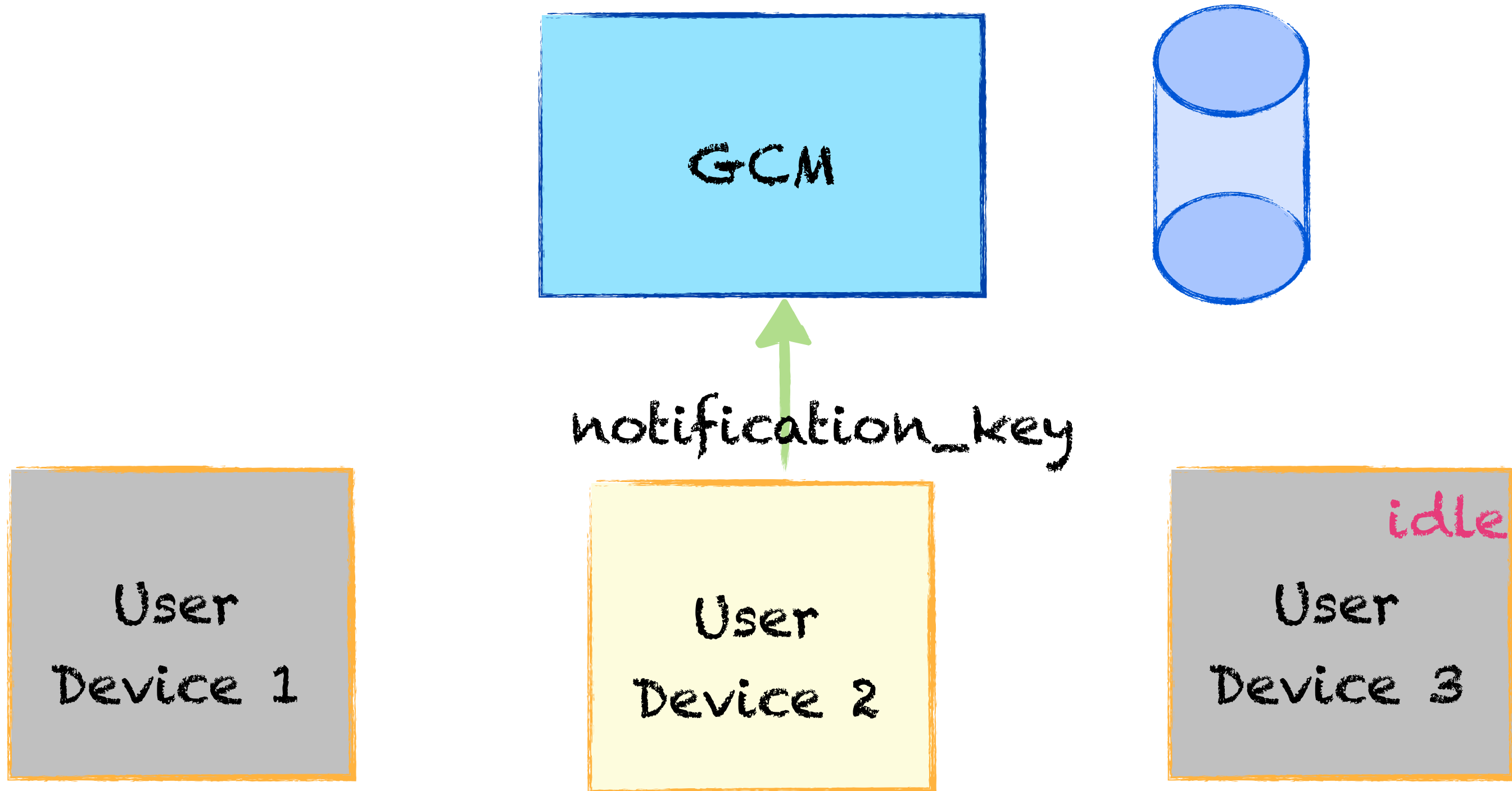
# Google Cloud Messaging Optimization



# Google Cloud Messaging Optimization



# Google Cloud Messaging Optimization



# Google Cloud Messaging Notification Key

Max 10 regIds per Notification Key



# Google Cloud Messaging Notification Key



Max 10 regIds per Notification Key



# Google Cloud Messaging Notification Key



Max 10 regIds per Notification Key



Notification Key Name is unique





# Google Cloud Messaging Notification Key



Max 10 regIds per Notification Key



Notification Key Name is unique



Available for HTTP and XMPP









<https://developer.android.com/google/gcm>



Check out the new docs!



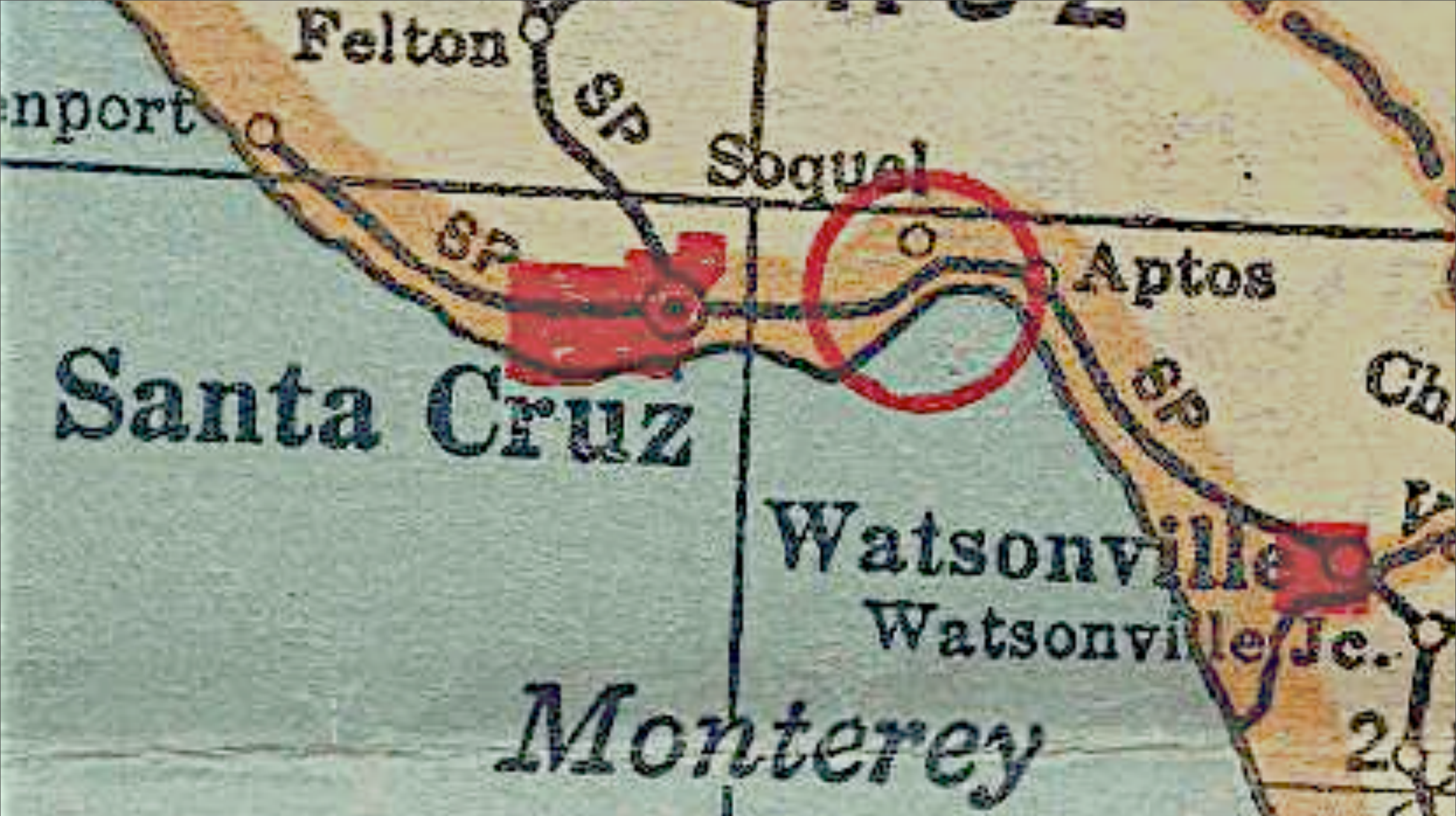
<https://services.google.com/fb/forms/gcm>



Sign Up to try out the new  
features !!!









Thanks!

[plus.google.com/  
+FrancescoNerieri](https://plus.google.com/+FrancescoNerieri)