NOTIFICATIONS WITH NATIVE MOBILE APPLICATIONS

THPV011

B. Bertrand*, J. Forsberg, MAX IV, Lund, Sweden
E. Laface, G.Weiss, European Spallation Source ERIC, Lund, Sweden

EUROPEAN SPALLATION SOURCE

ICALEPCS 2021

Notify server

FastAPI + PostgreSQL

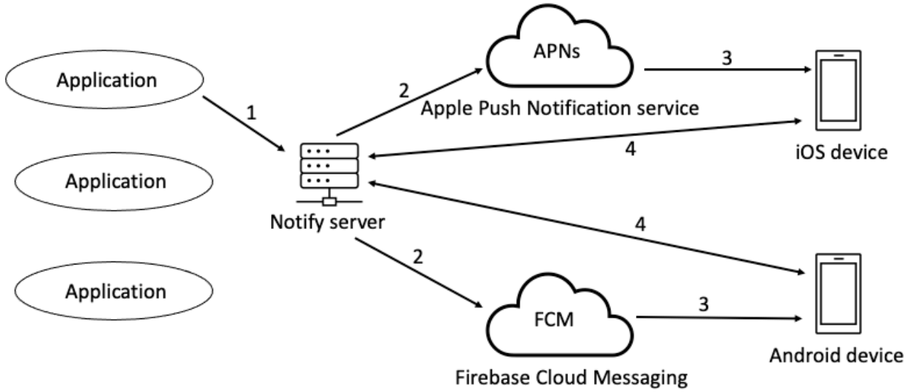
REST API
- Receive notifications
- Communicate with mobile clients

Sending notifications

POST to /services/{service_id}/notifications

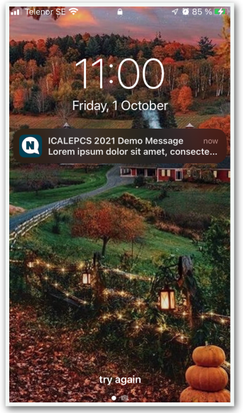
A **service** is a category to group notifications.

Notification workflow

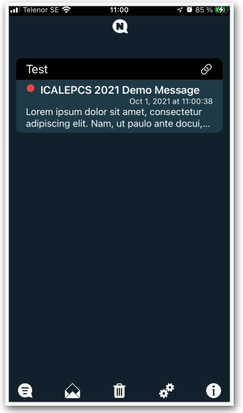


Notification workflow

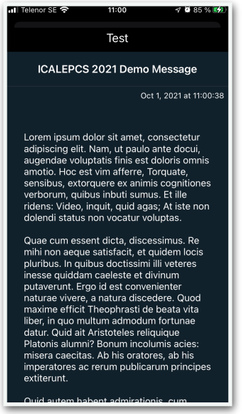
iOS and Android clients



Push notification



Messages list



Single message view

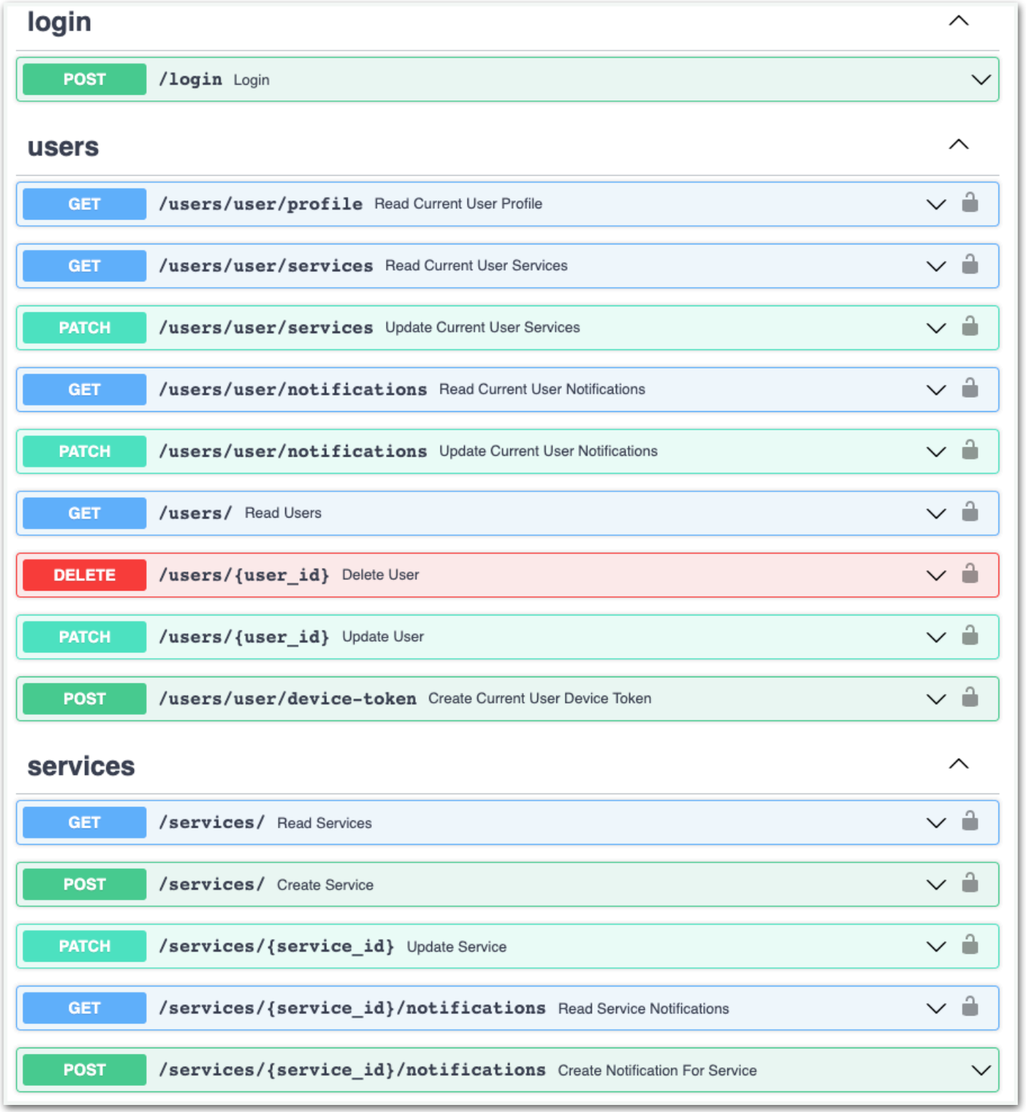
Notify Server

FastAPI + PostgreSQL

- Async Python web framework
- Based on Starlette, a lightweight ASGI framework and Pydantic, a data validation library using python type annotations
- High performance
- Fast to code
- Based on OpenAPI standard
- API documentation via Swagger UI
- Postgresql as database

```python
@router.get("/", response_model=List[schemas.Service])
def read_services(
    db: Session = Depends(deps.get_db),
    current_user: models.User = Depends(deps.get_current_user),
):
    """Read all services"""
    db_services = crud.get_services(db)
    return db_services
```

read_services endpoint source code



Notify server API

Sending notifications

Send a **POST** to **/services/{service_id}/notifications** with the fields:

- title
- subtitle
- url

```
curl -X 'POST' \
  'https://notify.maxiv.lu.se/api/v2/services/f848b451-e3c8-497b-a6da-2cb571d9d95e/notifications' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "title": "My message",
  "subtitle": "This is a test",
  "url": ""
}'
```

Sending a notification with curl

```
data = {"title": title, "subtitle": subtitle, "url": url}
response = httpx.post(
    f"{server}/api/v1/services/{service_id}/notifications", json=data
)
```

Sending a notification from Python

A **service** is a category to group notifications.

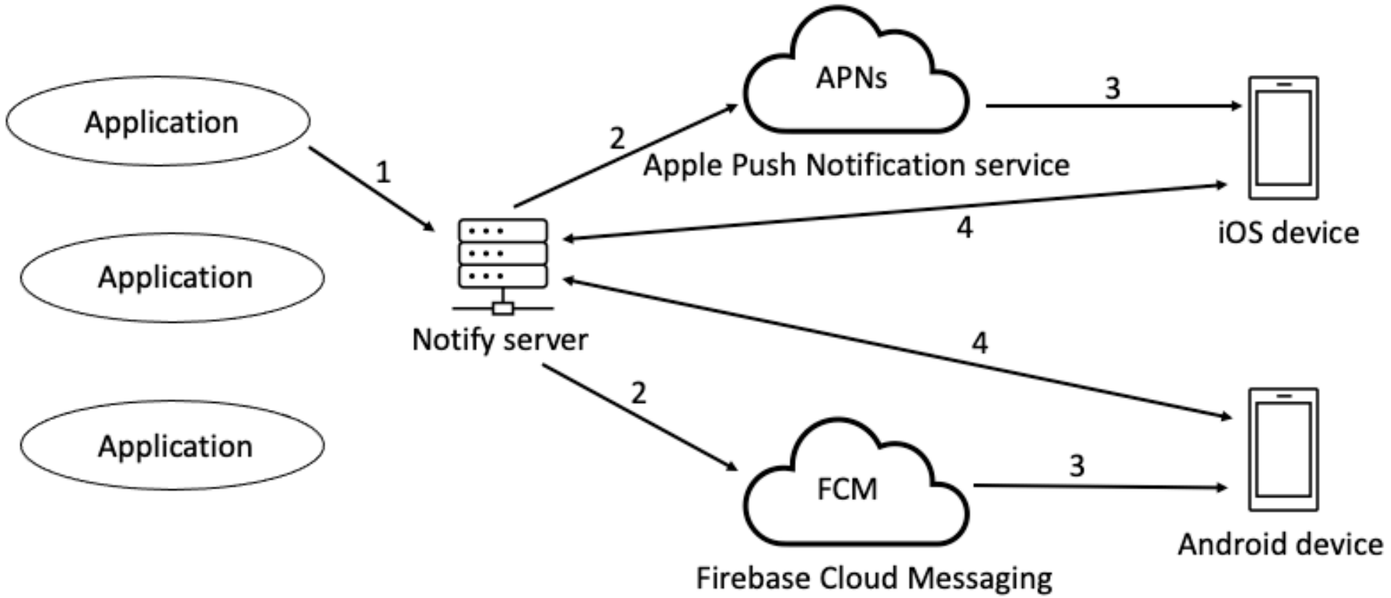
Each service is identified by a *service_id* (UUID).

Users can subscribe to the service they want.

Integrated in:

- ESS LogBook
- OpenXAL
- EPICS alarms via Kafka
- Prometheus
- Achtung (new alarm management system for Tango)

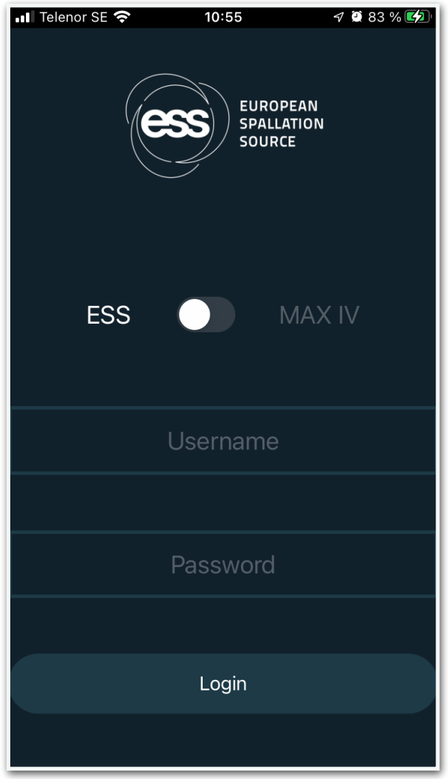
Notification workflow



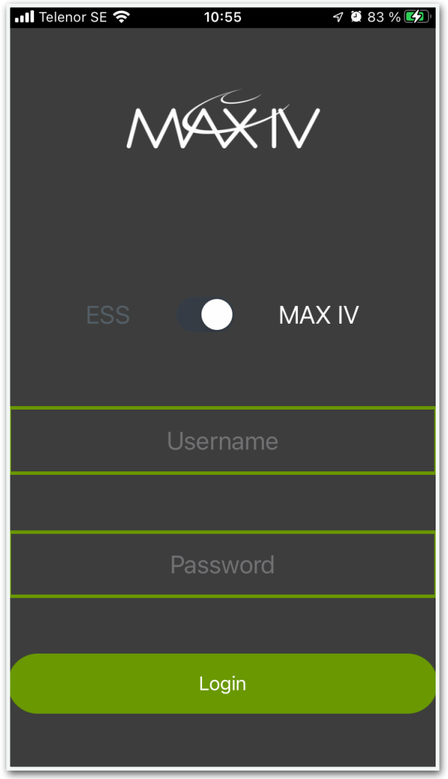
Notification workflow

1. An application sends a message to the Notify Server (the message is linked to a service).
2. The server sends a notification to both Apple Push Notification service and Firebase Cloud Messaging (depending on the device token type) for all users who subscribed to that service.
3. The notification is sent by Apple and Google cloud services to the user device.
4. When the user opens the notification, the full message is read from the Notify server API.

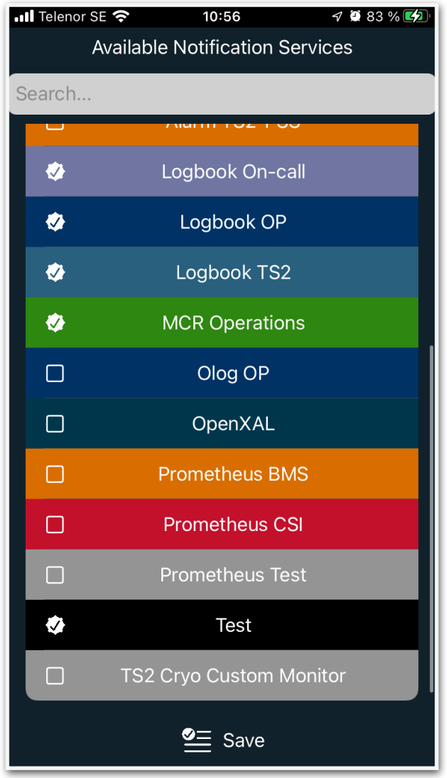
iOS and Android clients



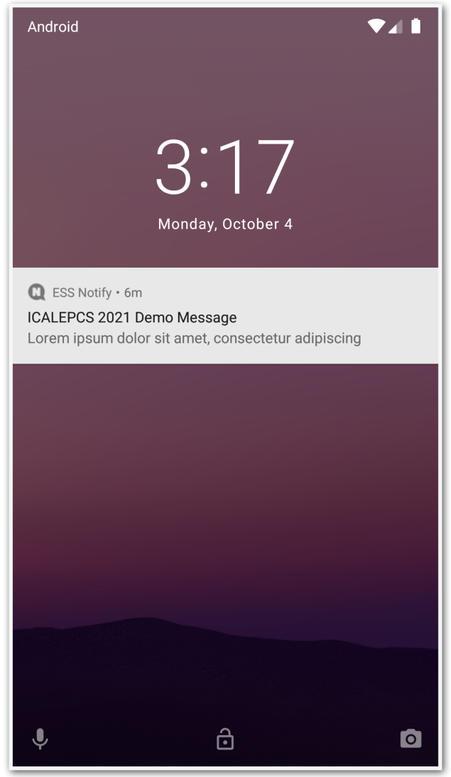
ESS Login screen (iOS)



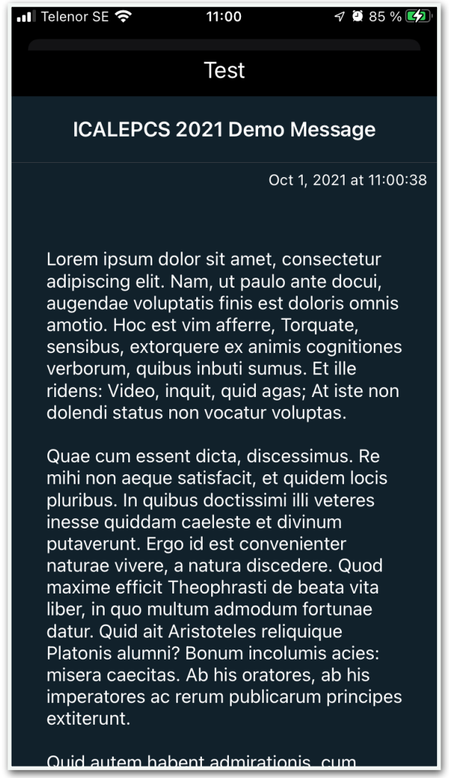
MAX IV Login screen (iOS)



List of Services (iOS)



Push notification alert (Android)



Single message view (iOS)