Mobile design patterns for authenticated users

fitbit

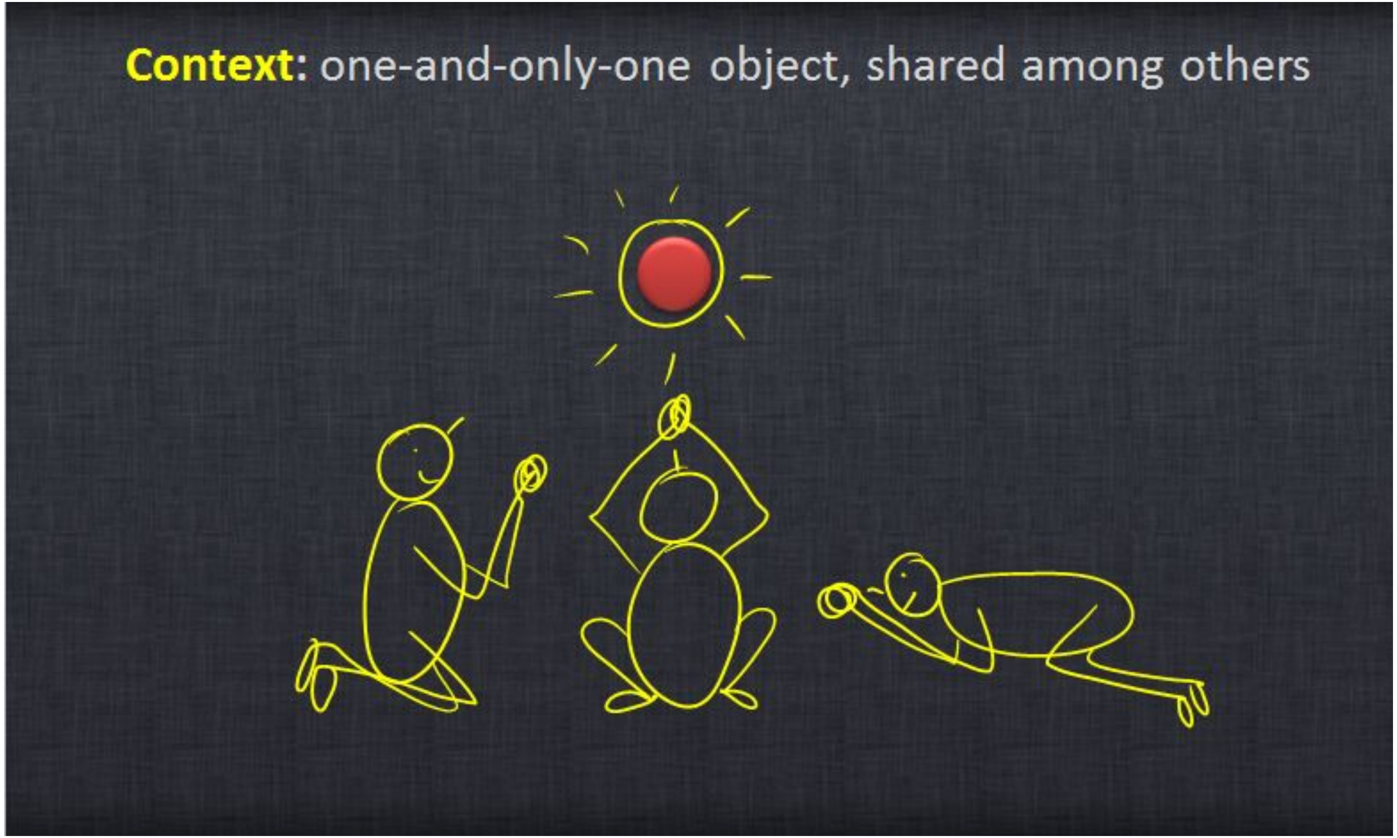
#define “Timofticiuc Andrei” timo

- Senior software engineer
- Infrastructure team
- All things network, data storage, sessions
- Cats and black metal

fitbit

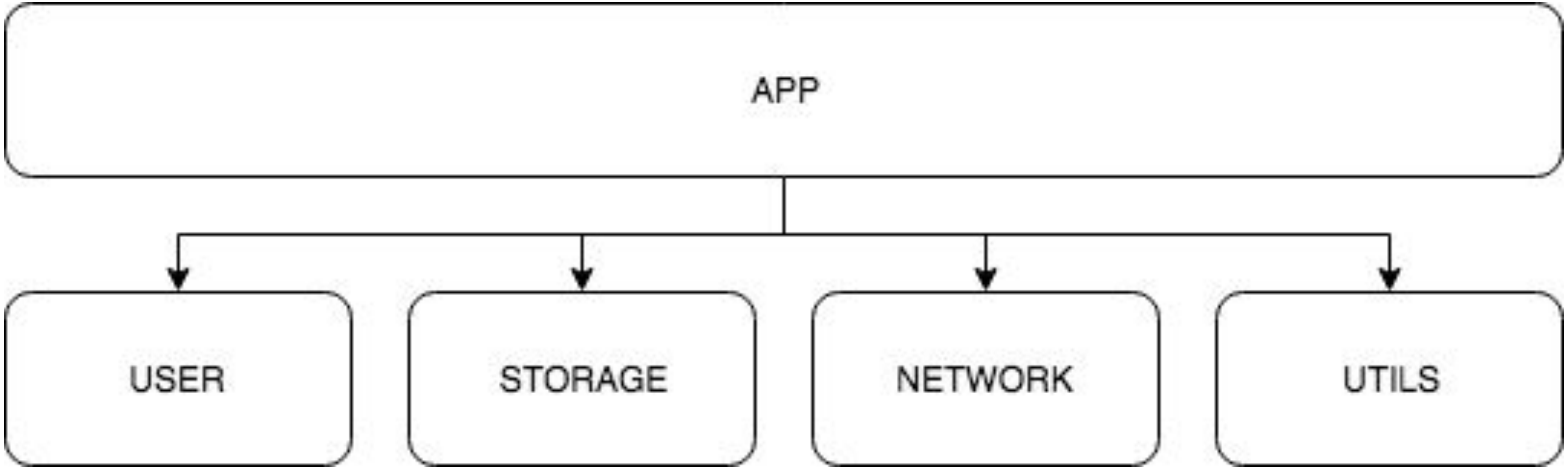
How not to handle ‘authenticated user’-apps

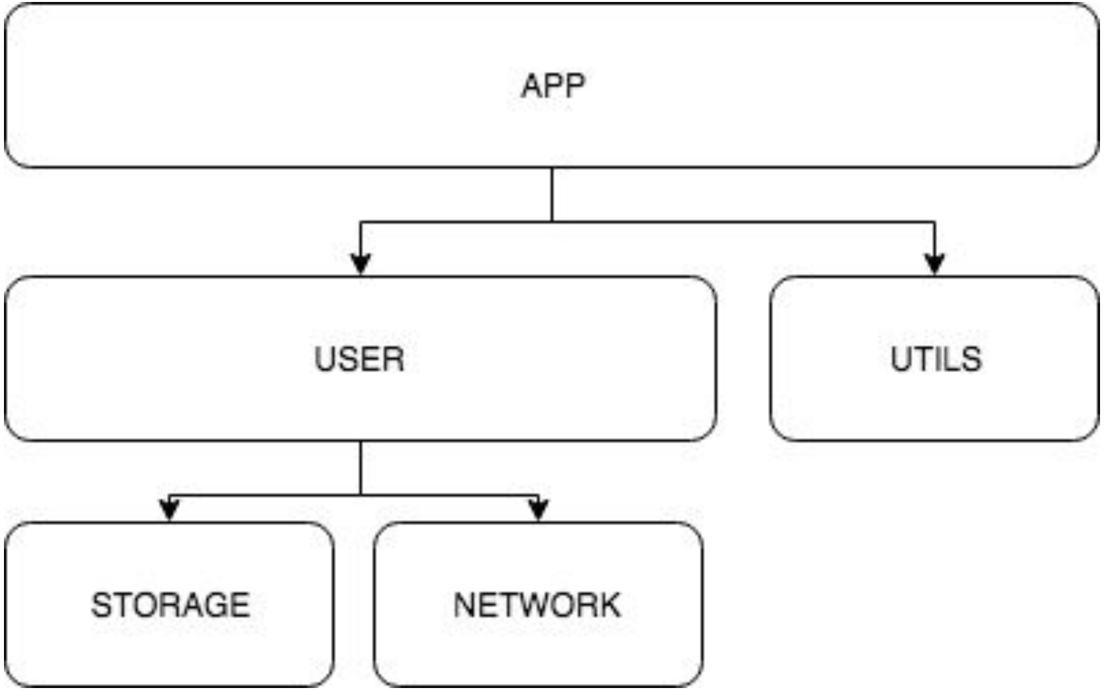
- All your stuff gets loaded after the user
- Don’t ever assume he’ll be logged in
- He doesn’t need to know all the details
- Fail as gracefully as a russian ballerina



Context: one-and-only-one object, shared among others

fitbit


APP
USER
STORAGE
NETWORK
UTILS


APP
USER
UTILS
STORAGE
NETWORK

PROTIP: Background time

- Don’t overstay your welcome (30s max)
- Don’t try to trick the system
- **NO UI stuff**
- Always attempt fallbacks if possible

OAuth2

- Login, refresh, repeat until 400
- acces_token, refresh_token, expires_in
- All authenticated requests get signed with valid authorization - no exceptions

HTTP status codes that matter

- 200: Great success
- 401: Expired token
- 409: Refreshing too fast
- 400: S#!t’s f#(k3d, logout
- 500: ¯_(ツ)_/¯
- Default: try again

fitbit

OAuth2 + Background = Love, Hate, Love

- Issue refresh as soon as it’s apparent
- Suspend everything else
- Save token and resume as fast as possible
- Failure to do so:
    - Negotiate with server
    - Have the request made on your behalf

TL;DR

- Don’t load useless stuff if user isn't logged in
- Be extra careful what you do in background
- Refresh often
- Logout is only a final measure

Q&A

fitbit