

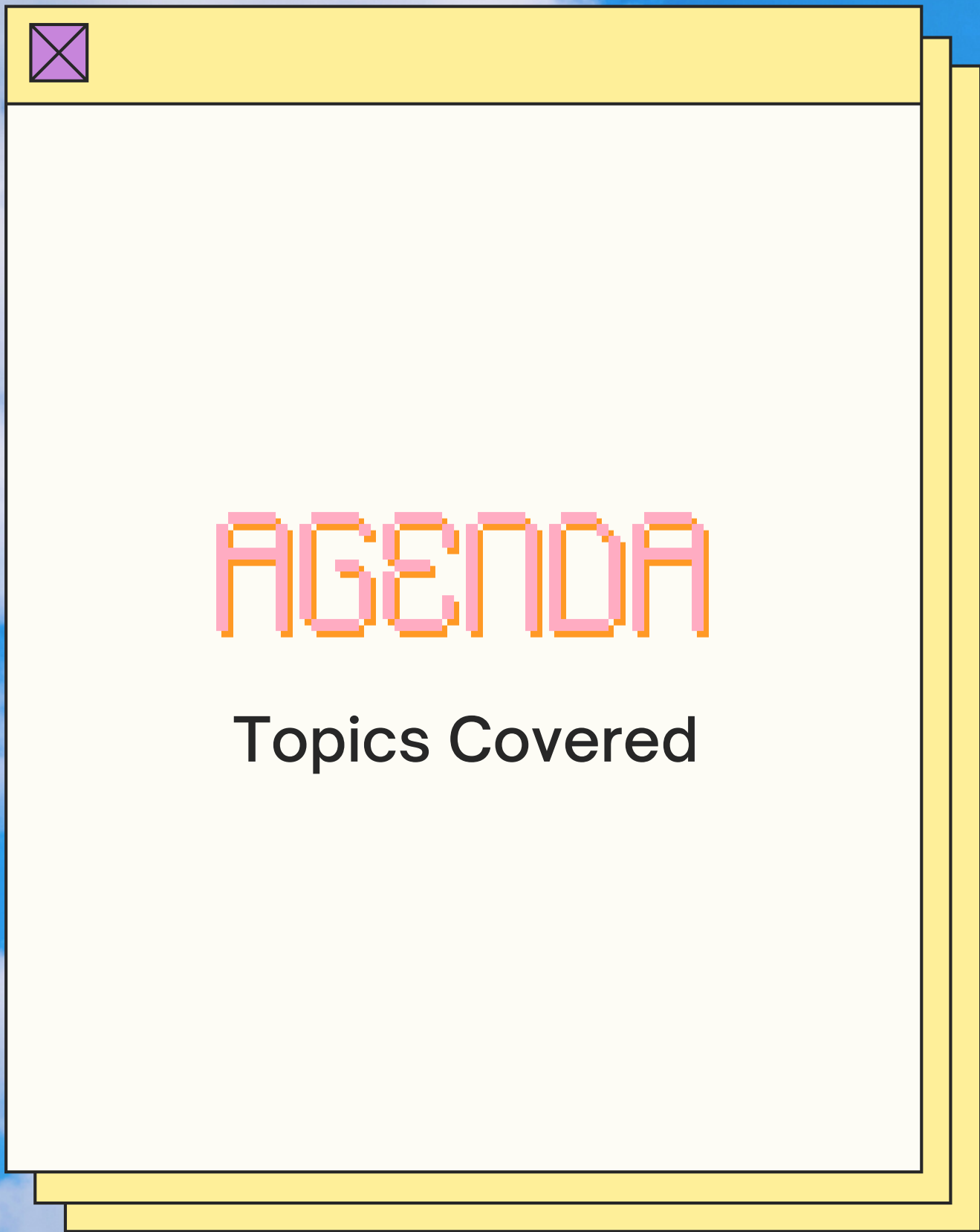
3. GRUPPETIME
MVVM, VITSEAPP OG
API-KALL



RELEVANT TIL OBLIG 2!



MENTI

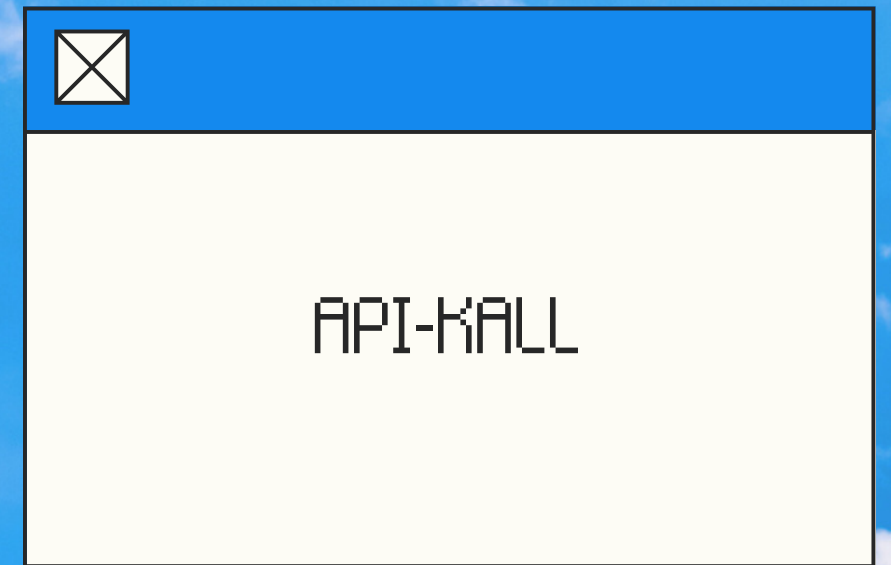


AGENDA

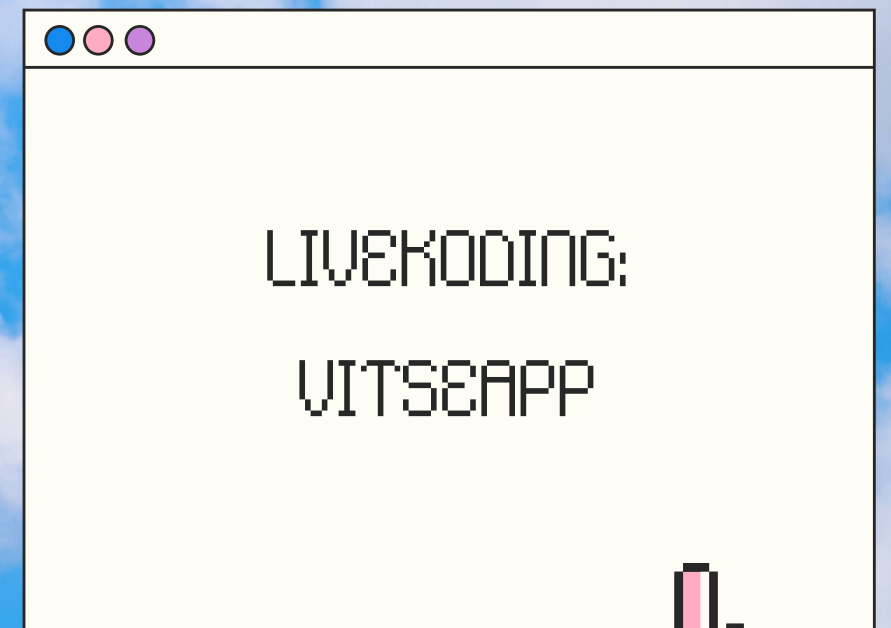
Topics Covered



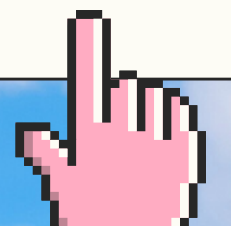
mvvm



API-KALL



LIVEKODING:
UITSEAPP



INNLEDNING

HUSK PÅMELDING TIL KICKOFF

FRIST 11. FEBRUAR

Når: Lørdag 17. februar fra kl. 12.15
- 16.00

HVORDAN LEVERE ET
PROSJEKT



HVA KAN DU OM MVVM?

ARKITEKTUR



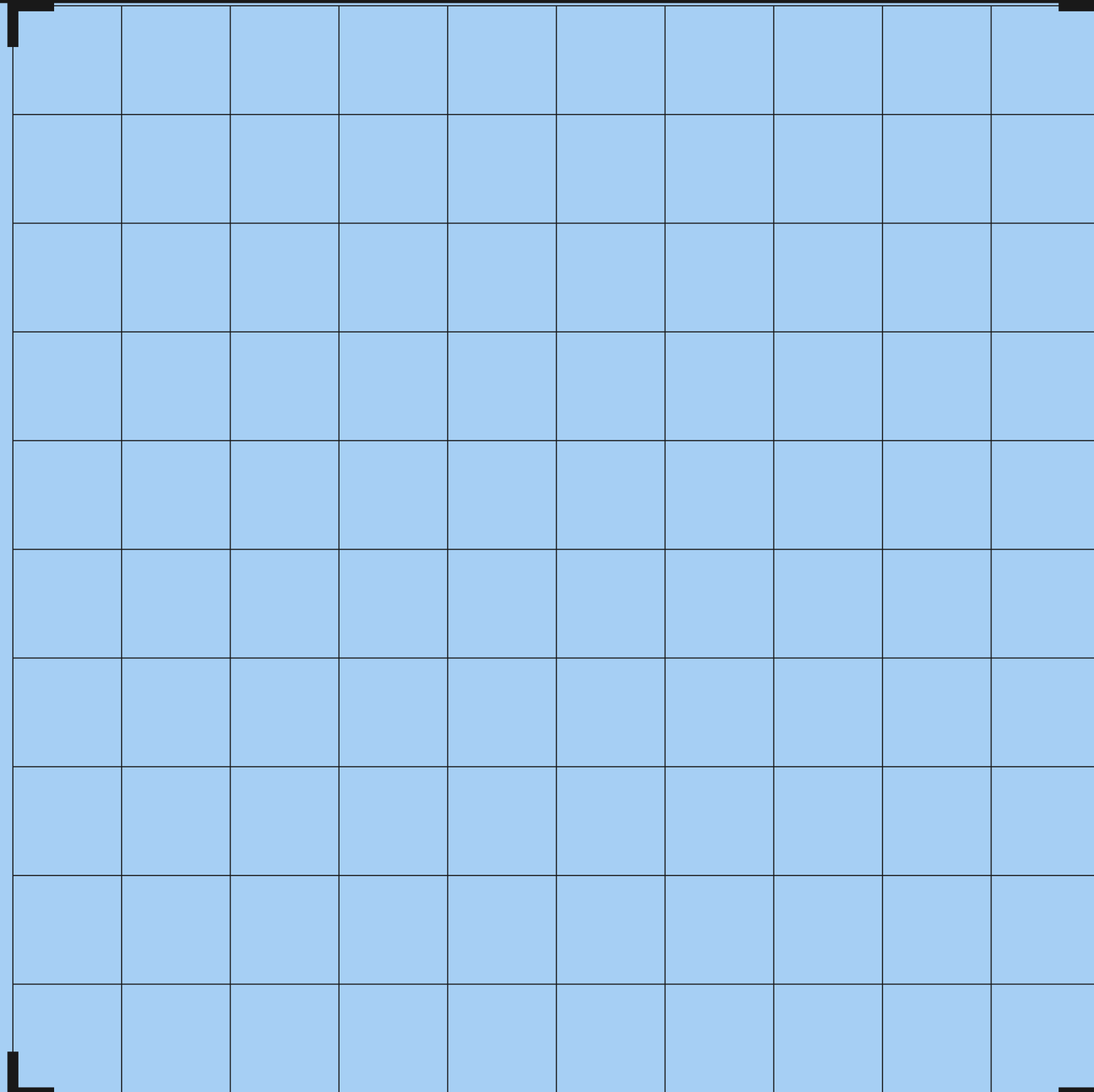
Hvorfor burde arkitektur være på plass før man starter å programmere?

I **startfasen**, så hjelper det å visualisere og tenke over hvordan appen skal fungere:

- Hva skal appen gjøre?
- Hvilke type data har vi / trenger vi?
- Hvordan skal vi representere dataen?
- Må vi gjøre noe med dataen?

Under **utviklingsfasen**:

- Lettere å vedlikeholde, skalere og teste appen
- Endringer i arkitektur i store prosjekter er tidskrevende og dyrt => teknisk gjeld

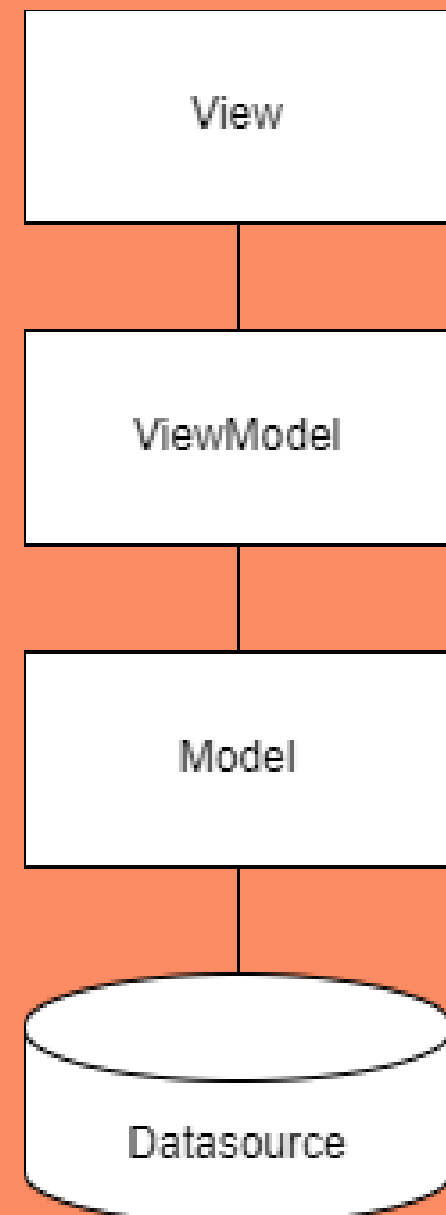


MVVM => MODEL, VIEW, VIEWMODEL

Standard arkitektur i Android

Poenget?

- Klart skille mellom User Interface (UI) og appens data



UI laget:

Ansvarer til **View**:

- Presentere state til bruker (ved å observere state i ViewModel)
- Muliggjøre brukerinteraksjon

Ansvarer til **ViewModel**:

- Presentere state til View-et
- Oppdatere state
 - Henting av data
 - Reagere på brukerinput

Data laget:

Ansvarer til **Model**:

- Hente data
 - Gjøre et API-kall
 - Behandle dataen
- Presentere ferdig behandlet data til ViewModel

TIPS TIL MVVM



[Back to Agenda Page](#)



Mappe-struktur er **ikke** lik
arkitektur



Behandling av hentet data skal
kun skje i datalaget

A partial grid of empty cells, likely representing a data table or layout structure.

HVORFOR ER MVUM
VIKTIG?

HVA KAN DU OM API?

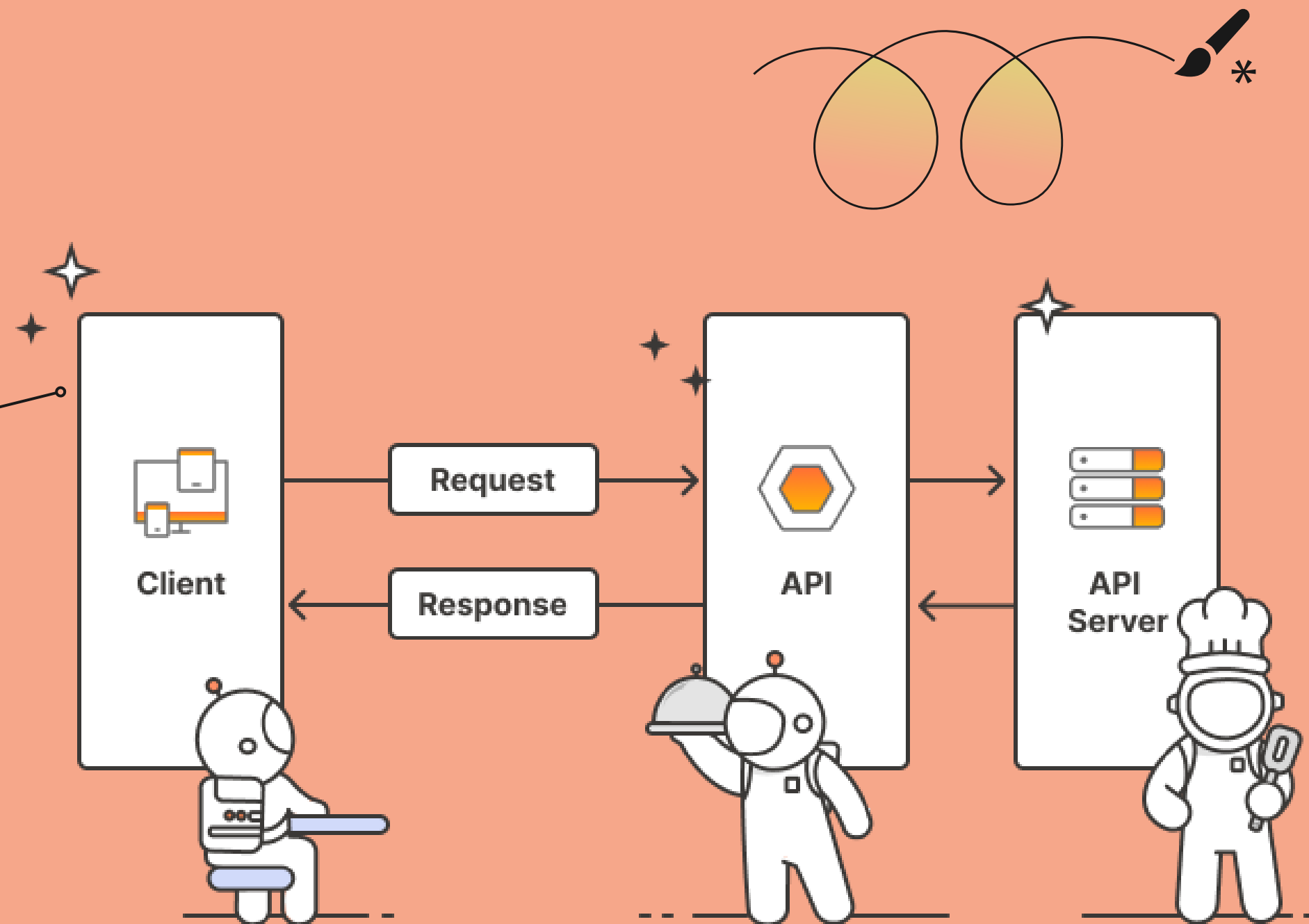
API



[Back to Agenda Page](#)

Application **P**rogramming **I**nterface

- Kan tenkes på som en meny på en restaurant
 - Liste over retter, m. beskrivelse
 - Bestiller fra kjøkkenet
 - Leverer akkurat riktig rett i henhold til bestillingen





REST API

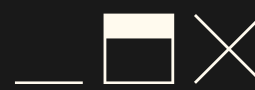


Representational State Transfer

Restfull API



- Hente (GET)
- Oppdatere (PUT / PATCH)
- Slette (DELETE)
- Legge til (POST)



- Et sett med prinsipper for nettbaserte applikasjoner
- Bro eller mellommann mellom en klient (for eksempel en nettside eller mobilapp) og en server, og lar klienten utføre operasjoner som å hente, oppdatere, slette eller legge til data på en server.

HVORFOR BRUKE APIER?



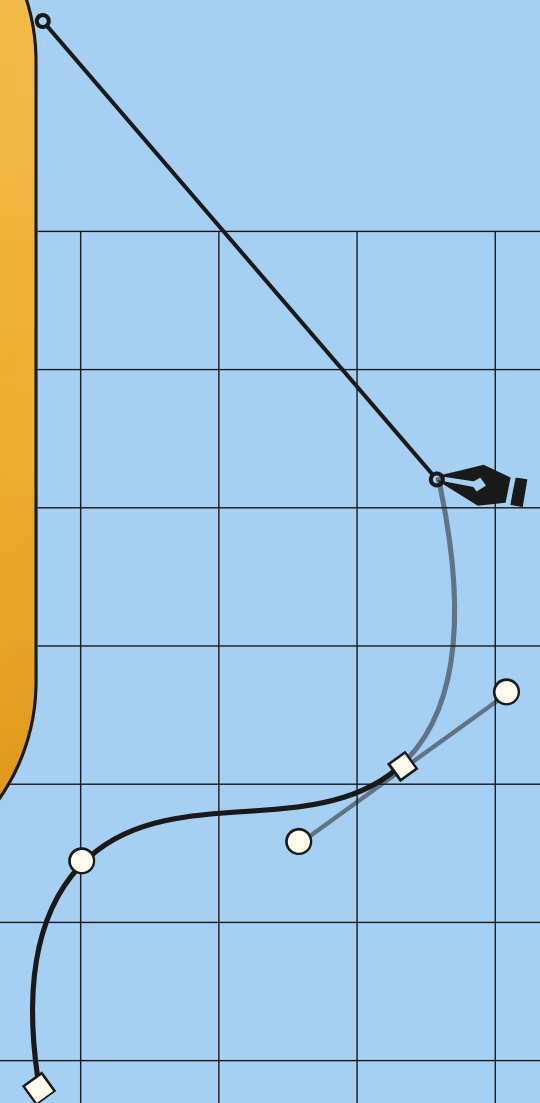
Search

- Forenkler utvikling
- Gjenbruk av kode
- Modularitet
- Integrasjon
- Innovasjon
- Sikkerhet



?

SAVE



STATUS KODER

1xx Informational

2xx Success

3xx Redirection

4xx Client error

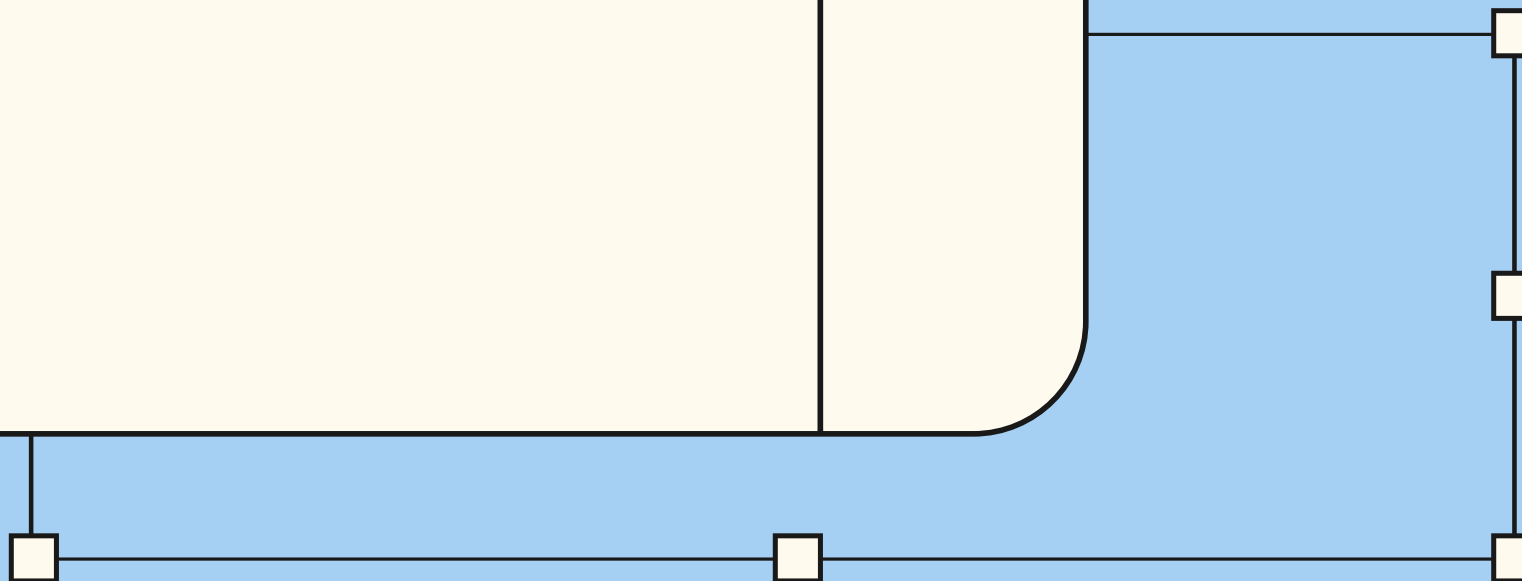
5xx Server error

100	Continue
200	OK
301	Moved Permanently
404	Not Found
503	Service Unavailable



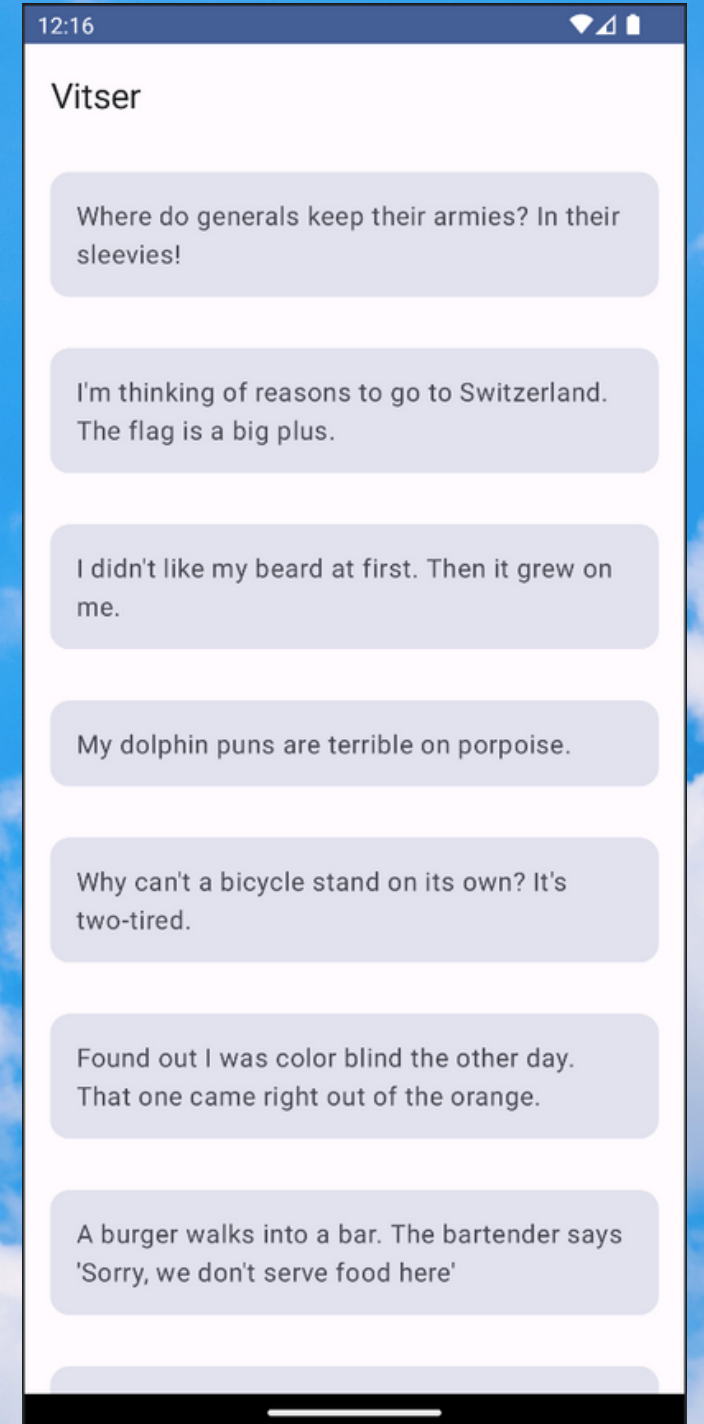


PAUSE

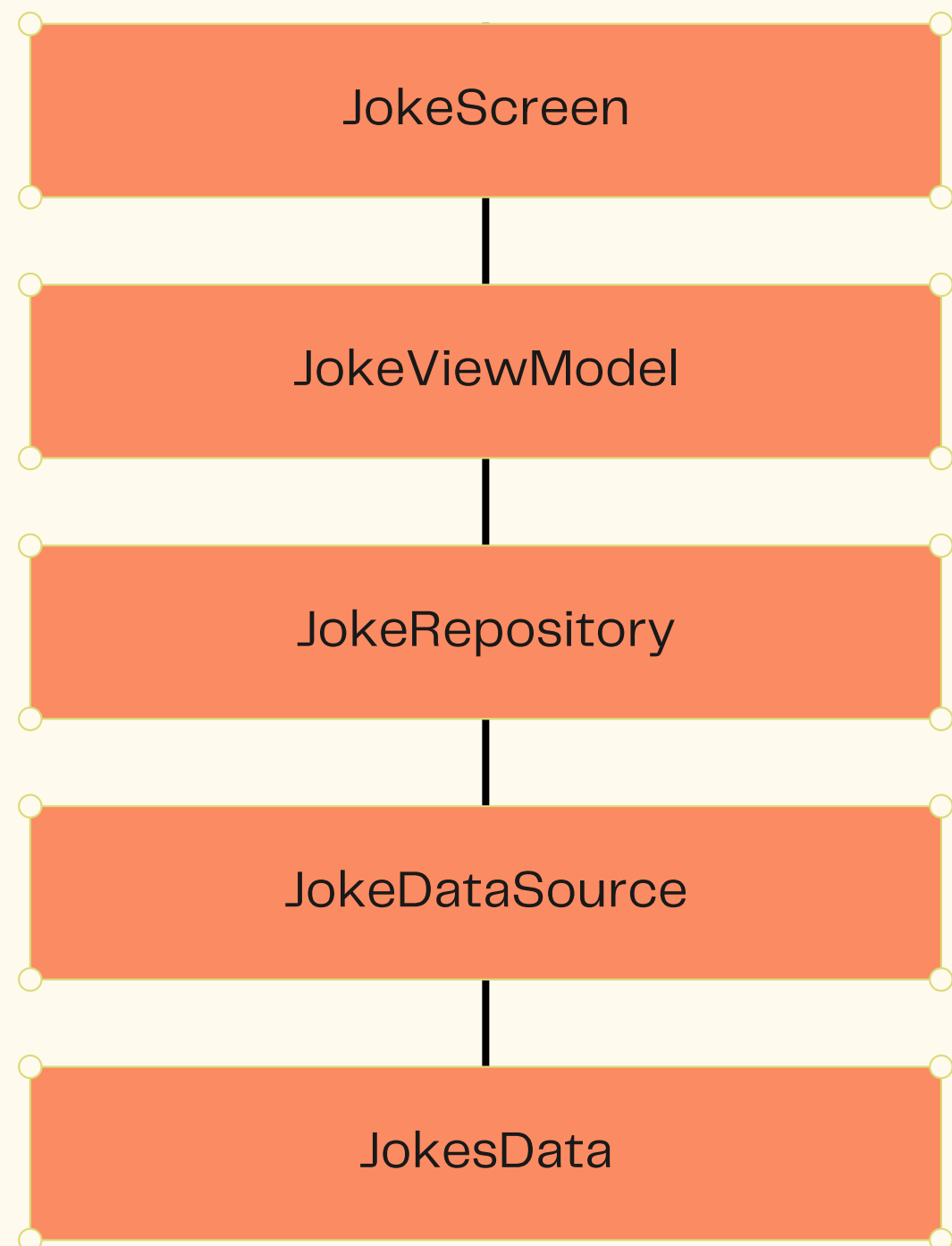


LIVEKNOODING:

VITSERAPP



VITSEAPP STRUKTUR



JOKESCREEN

- Viser vitser på skjerm
- Bruker JokesViewModel for å få tilgang til dataene

JOKEVIEWMODEL

- Håndterer logikken og datahentingene av vitsene
- Etter å ha mottatt data fra JokesRepository, oppdaterer JokesViewModel UI-tilstanden (jokeUiState), som igjen varsler JokeScreen om å oppdatere seg.

JOKEREPOSITORY

- Brukes til å kombinere datakilder

JOKEDATASOURCE

- Direkte interaksjon med ekstern datakilde (API)

JOKESDATA

- Enkel dataklasse som representerer en liste av vitser

API-KALL STEG FOR STEG



1. START MED Å LAGE DATAKLASSER SOM REPRESENTERER ENDEPUNKTET
2. LAG REPOSITORY
3. LEGGE TIL DEPENDENCIES I BUILD.GRADLE, OG INTERNETTILGANG I ANDROIDMANIFEST.XML
4. LAG JOKESDATASOURCE MED HTTP-KLIENT OG HTTP-FORESPØRSLER(EKS. GET). KJØR ASYNKRONT KALL
5. LAG JOKEVIEWMODEL
6. SCREENS

You got
this!

ENDEPUNKT

```
{
  "jokes": [
    {
      "id": "1",
      "joke": "Where do generals keep their armies? In their sleeves!"
    },
    {
      "id": "2",
      "joke": "I'm thinking of reasons to go to Switzerland. The flag is a big plus."
    },
    {
      "id": "3",
      "joke": "I didn't like my beard at first. Then it grew on me."
    },
    {
      "id": "4",
      "joke": "My dolphin puns are terrible on porpoise."
    },
    {
      "id": "5",
      "joke": "Why can't a bicycle stand on its own? It's two-tired."
    },
    {
      "id": "6",
      "joke": "Found out I was color blind the other day. That one came right out of the orange."
    }
  ]
}
```



```
data class Joke(
  val id: String,
  val joke: String
)
```

```
data class Jokes(
  val jokes: List<Joke>
)
```



TAKK FOR IDAG

Ingeborg - (ingebosos@uio.no)
Ole-Sander - (osjohans@uio.no)
Eirik - (eirikjl@uio.no)