

# Writing future proof Swift



Erwin Zwart  
iOS Developer @ The Capitals



# Code lives in an ever changing environment

How can you plan for the future if you don't know what's ahead?



# If time allows it

- Write automated tests
- Do code review

# On a code level

- Avoid assumptions, restrict your intentions
- Write code that models what it represent
- Write shy code

Avoid assumptions,  
restrict your intentions

*“The iPhone screen is always 320 x 460px”*

–A lot of people in 2012

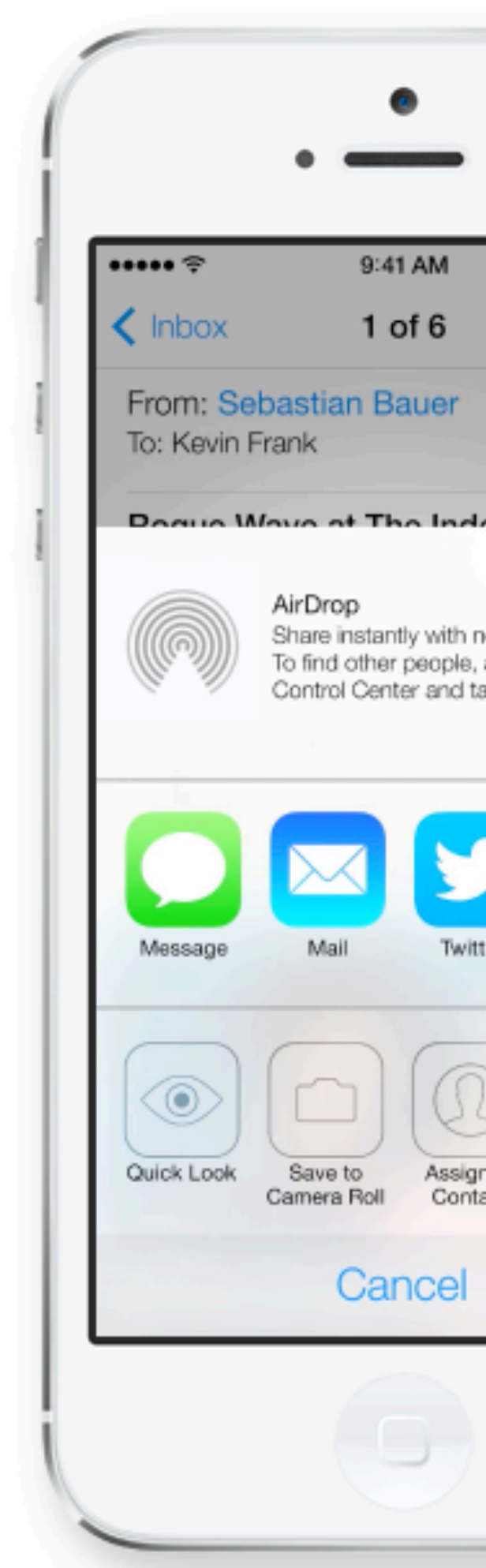
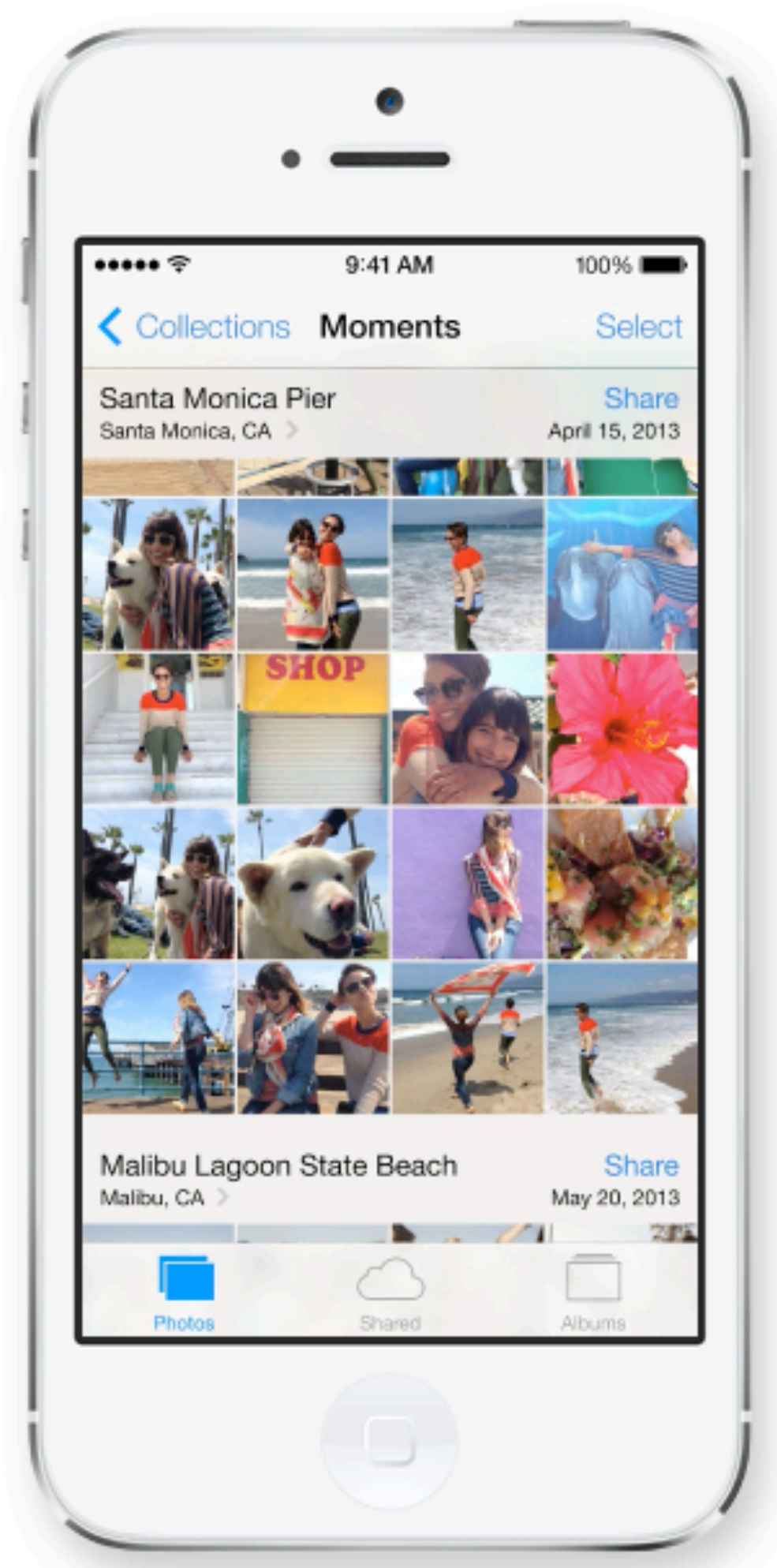
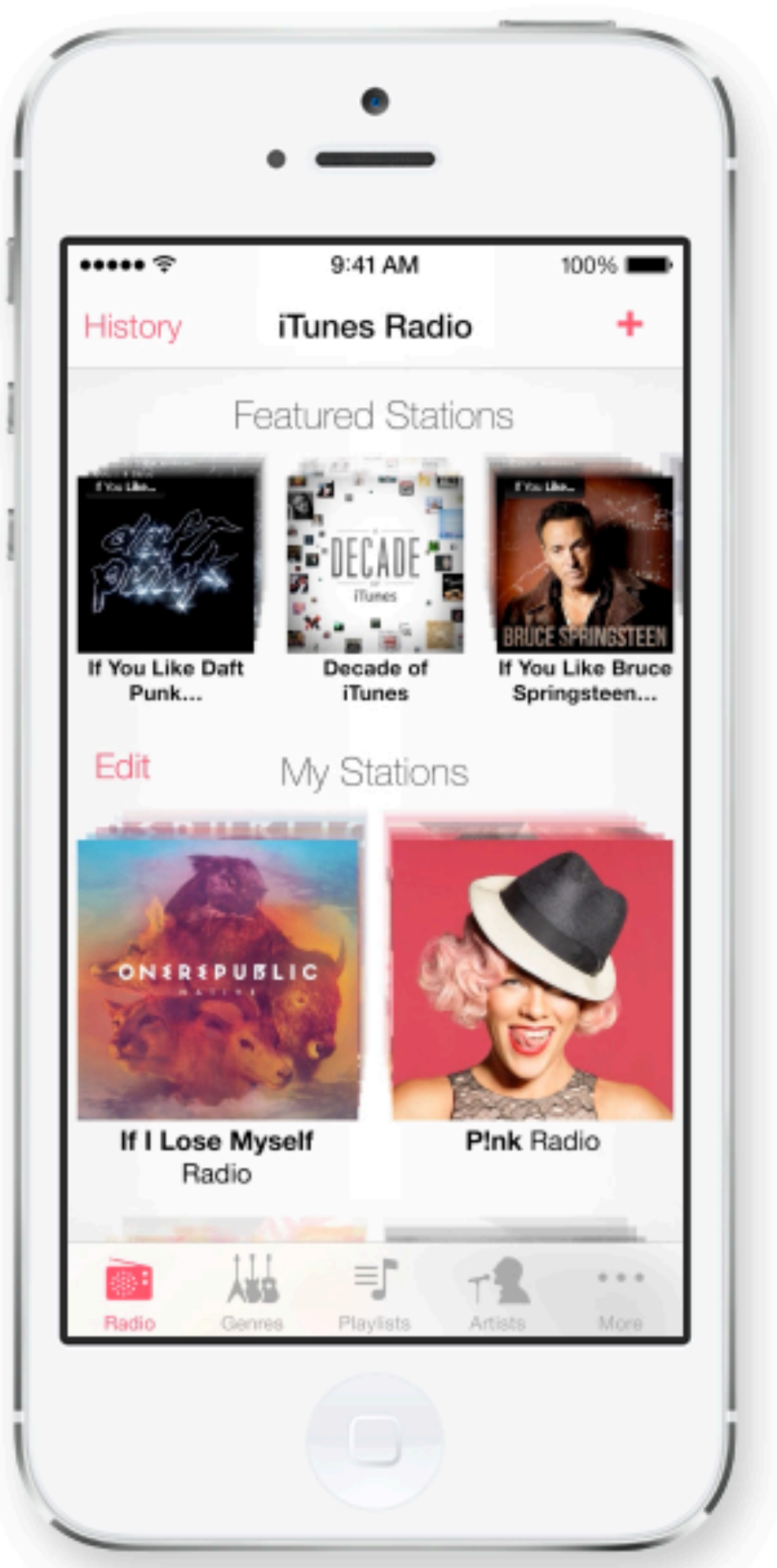




*“The navigation bar is always has a height of 64px”*

–A lot of people in 2014





Monday

10

# All of these assumptions are wrong

- There are always 24 hours in a day.
- Months have either 30 or 31 days.
- Years have 365 days.
- Any 24-hour period will always begin and end in the same day (or week, or month).
- A week always begins and ends in the same month.
- A week (or a month) always begins and ends in the same year.

Write code that models  
what it represent

# Write shy code

- Each unit should have only limited knowledge about other units: only units “closely” related to the current unit.
- Each unit should only talk to its friends; don't talk to strangers.
- Only talk to your immediate friends.



Examples

# Questions?



erwin@thecapitals.nl



@e\_zwart