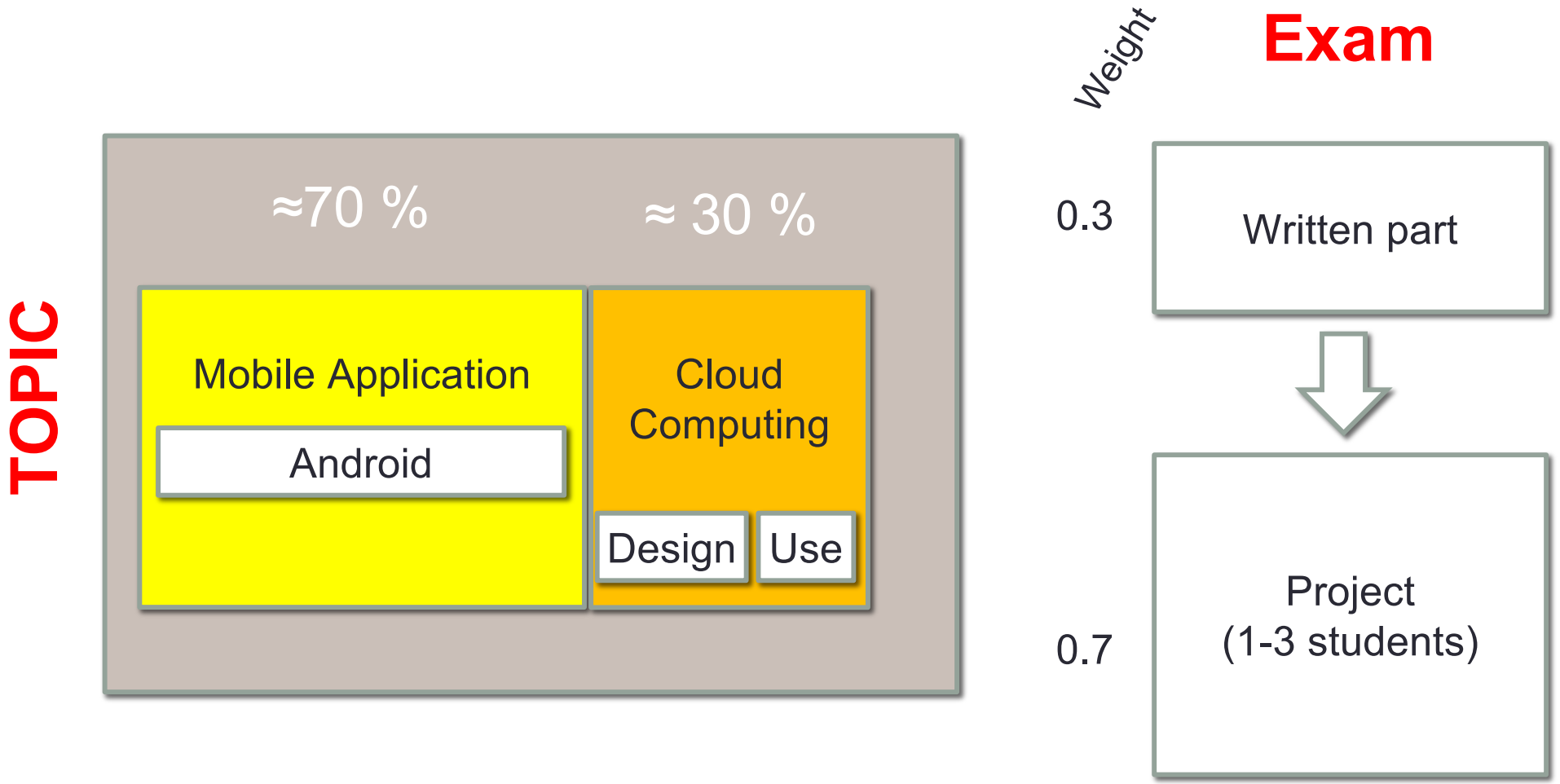
MOBILE APPLICATIONS AND CLOUD COMPUTING

Roberto Beraldi

Course Outline



Course homepage: *http://www.diag.uniroma1.it/~beraldi/MACC*

Materials

- There are not official books
- All the materials are slides ans pointers to internet recourses available on the web site of the course

Course outline

- Applications for Mobile devices
- The Android OS
- Android SDK Framework
- Activity,Fragment and Navigation
- Sensors and orientation
- 2D graphic
- 3D graphic (introduction)
- Storage
- Services, Brodcast receivers
- Cloud services

More about your project

service your app should use

- Multiuser
- Responsive UI
- Graphic
- Concurrency
- Sensors
- Data consistency

Cloud provider

- Authentication
- Web-api

Your cloud service

- CRUD interface
- Security aspects

Characteristics your application should have

Your cloud part

Why/How mobile apps differ from desktop one?

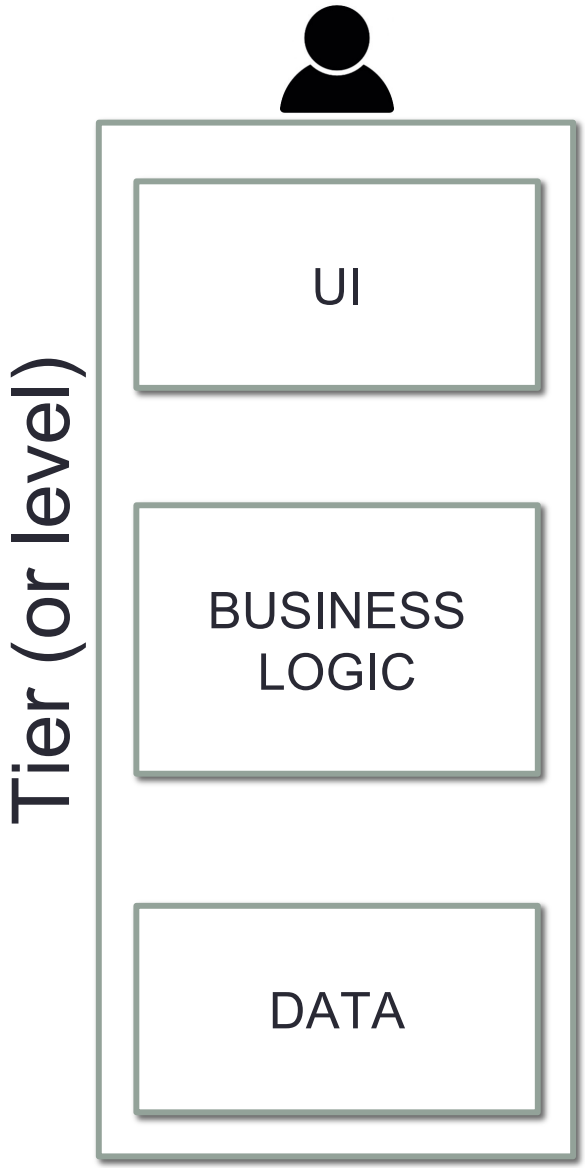
- Smaller Screen size
  - Organize the information accordingly, e.g., list
- Reacher Input/output system:
  - Input: sensors (GPS, orientation),..
  - output: vibration, ..
- Concern about energy consumption and efficiency
  - Apps have a lifecycle (can be suspended)
  - Memory management
  - Clock speed is tuned

User behaviour

- Very easy to install an application
  - Design UI is important
  - No Bugs (impossible) → TESTING
  - Crash (no second change)
  - Bad UX (User eXperience)
  - Slow

- Average number of installed applications ≅ 35
- Percentage of apps used per day ≅ 26 %

Functional decomposition of a generic app

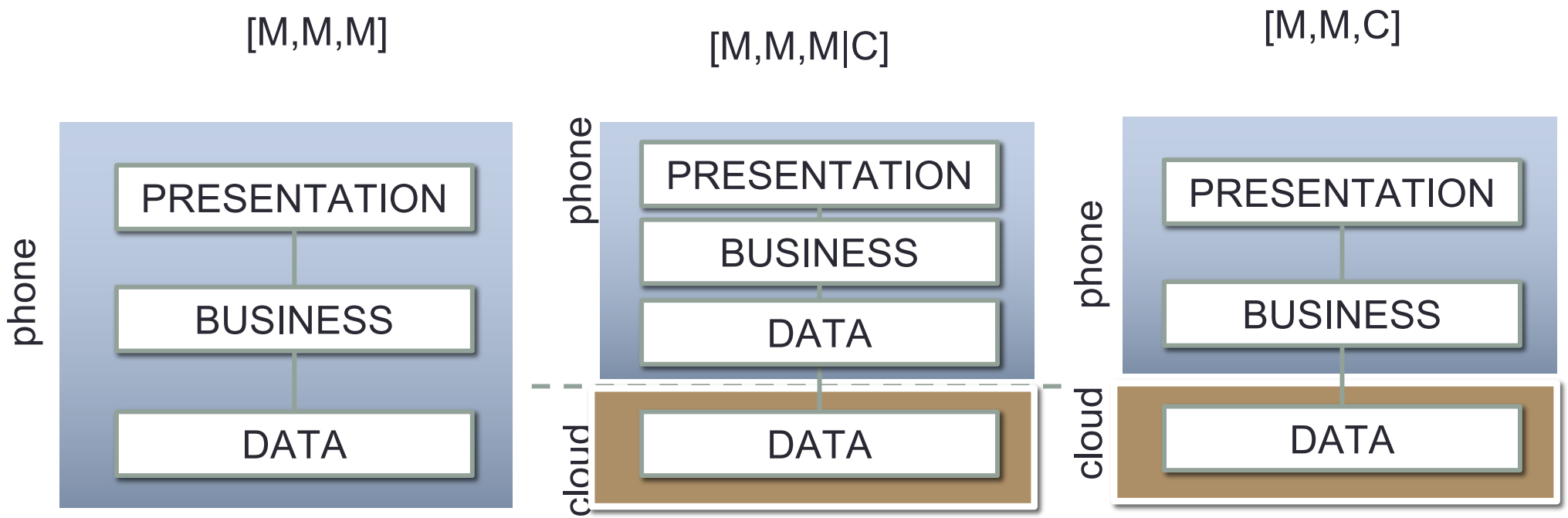


- User Interaction:
  - occurs mostly through GUI (gestures, …)
  - or other channels (vibration, notification)
- What the application ‘does’
- How data are stored and managed

Deploy architecture: how to map a three-tier architecture (tier→device)

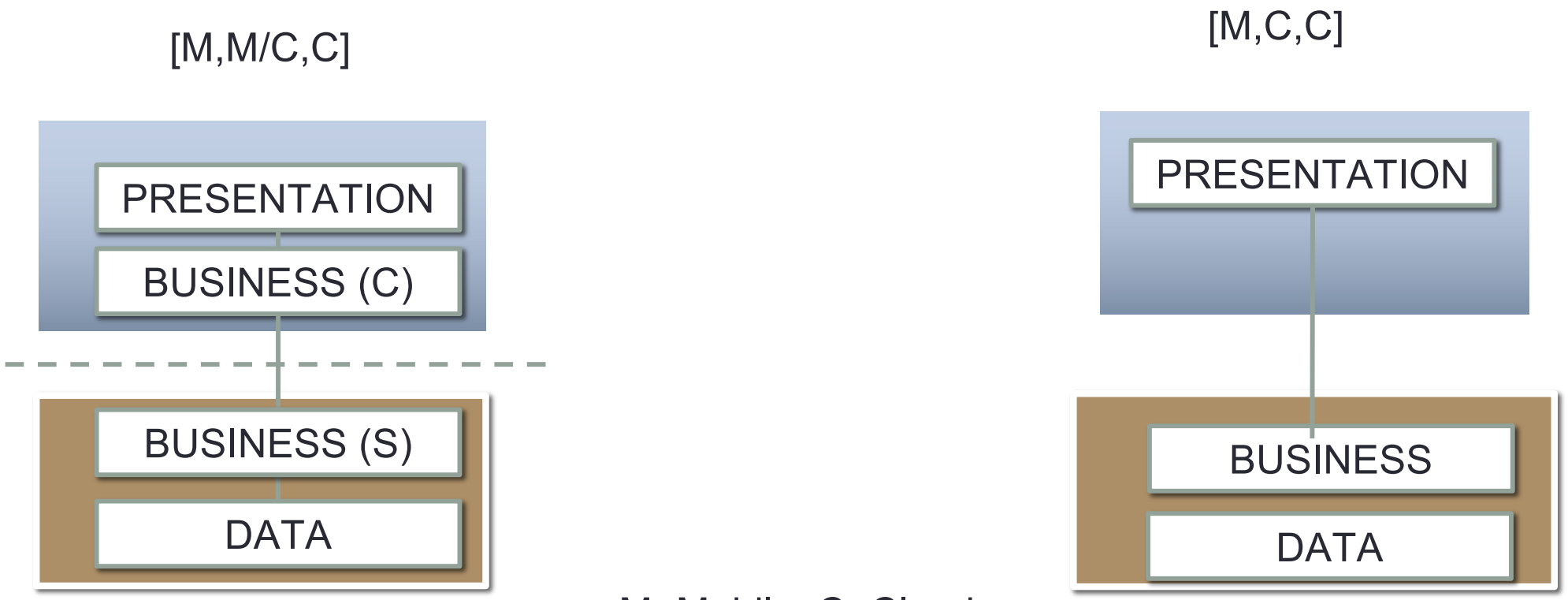
- Presentation tier → {M}
- Business logic → {M,C}
- Data tier → {M,C}

M=Mobile, C=Cloud

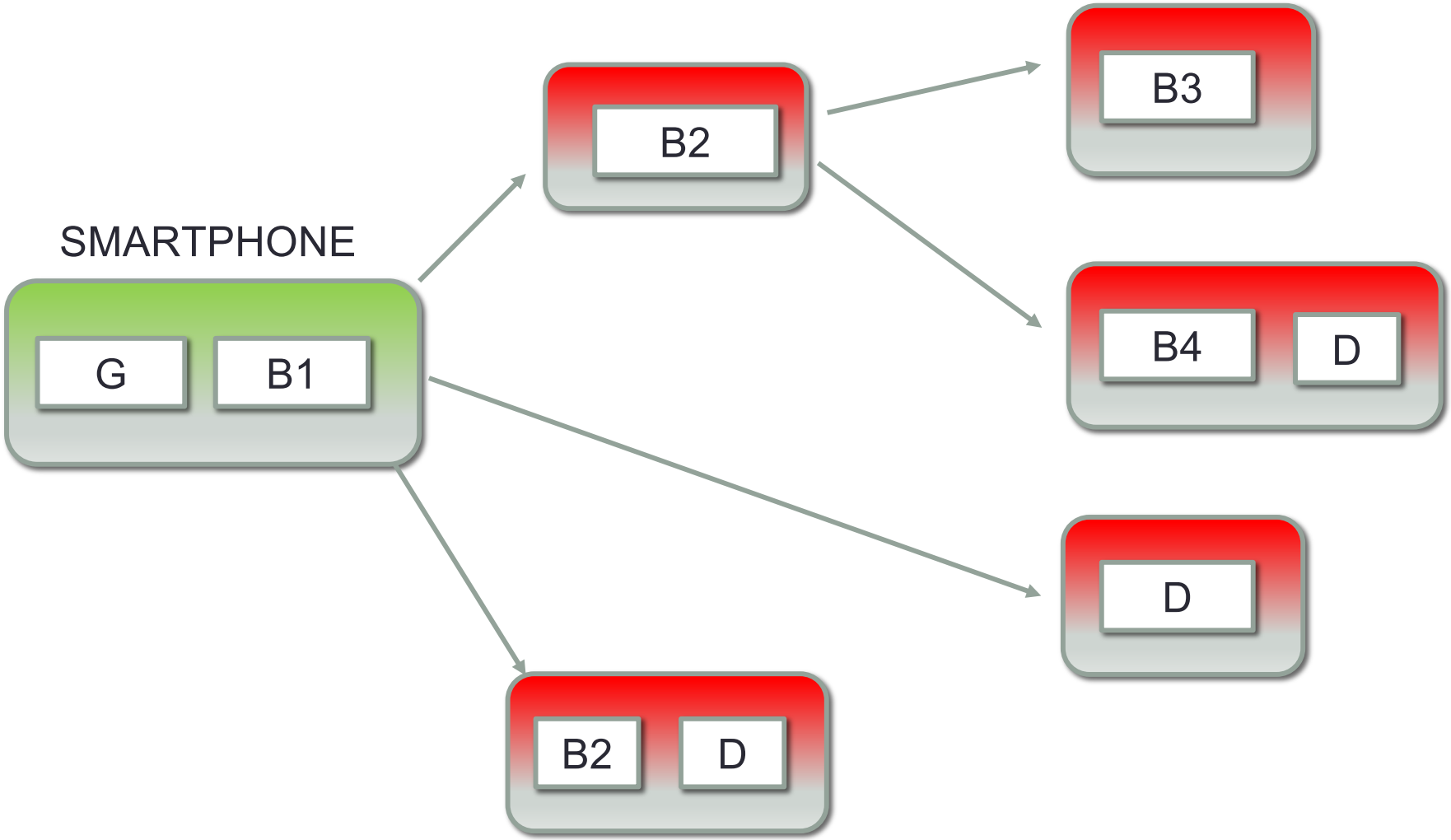


Deploy architecture: how to map a three-tier architecture (tier→device)

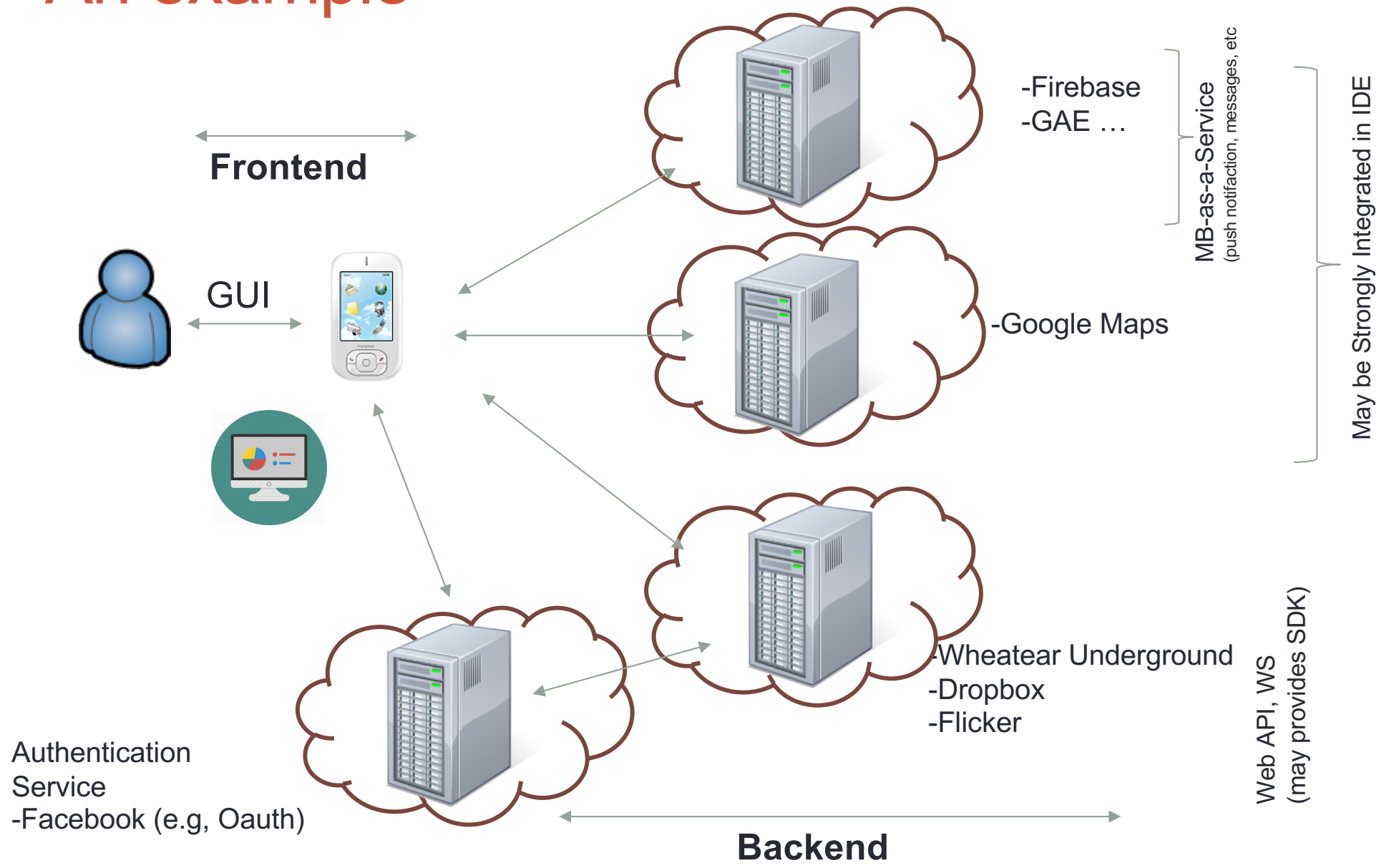
- Apps may strongly rely on a set of ‘services’ running remotely



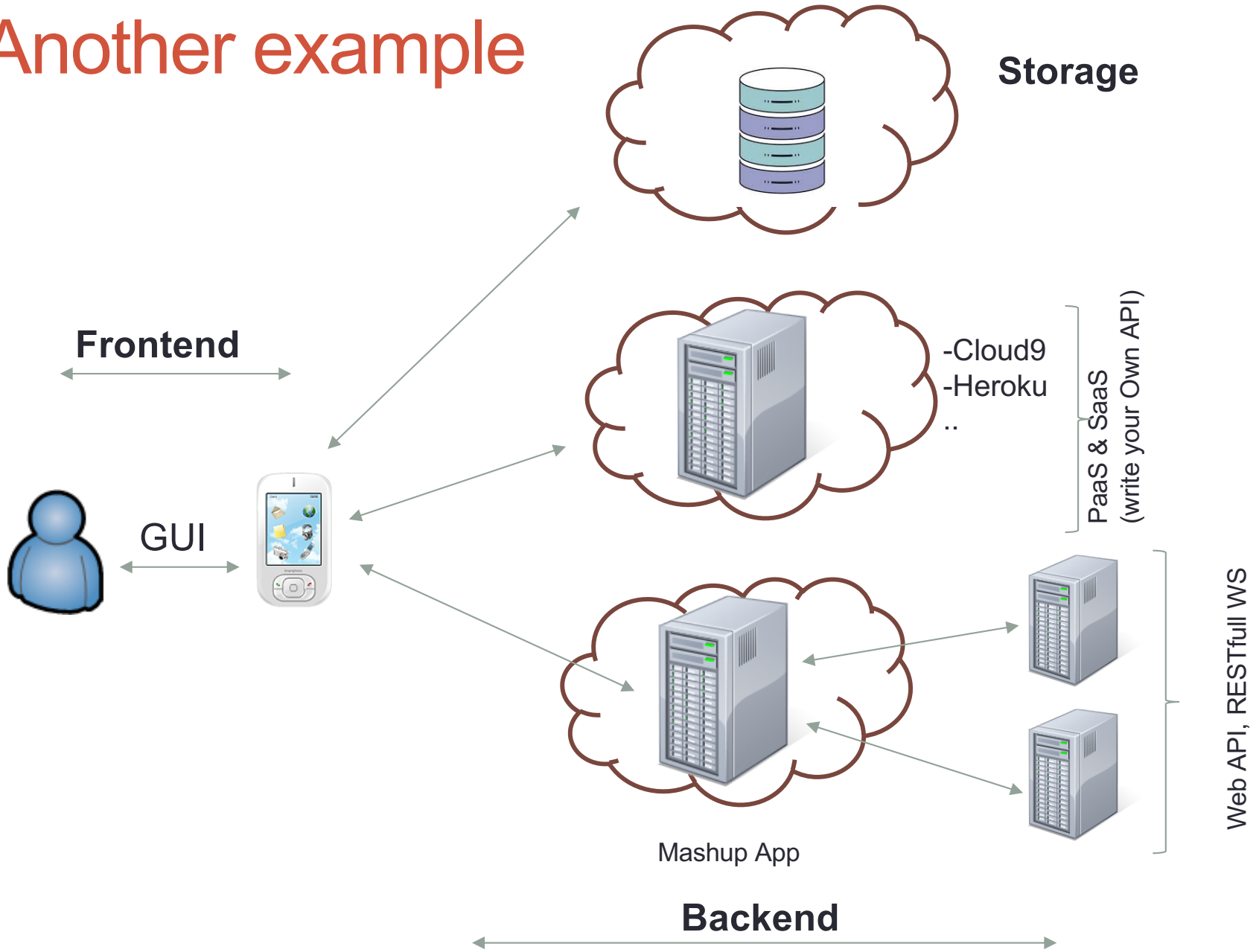
Mashup



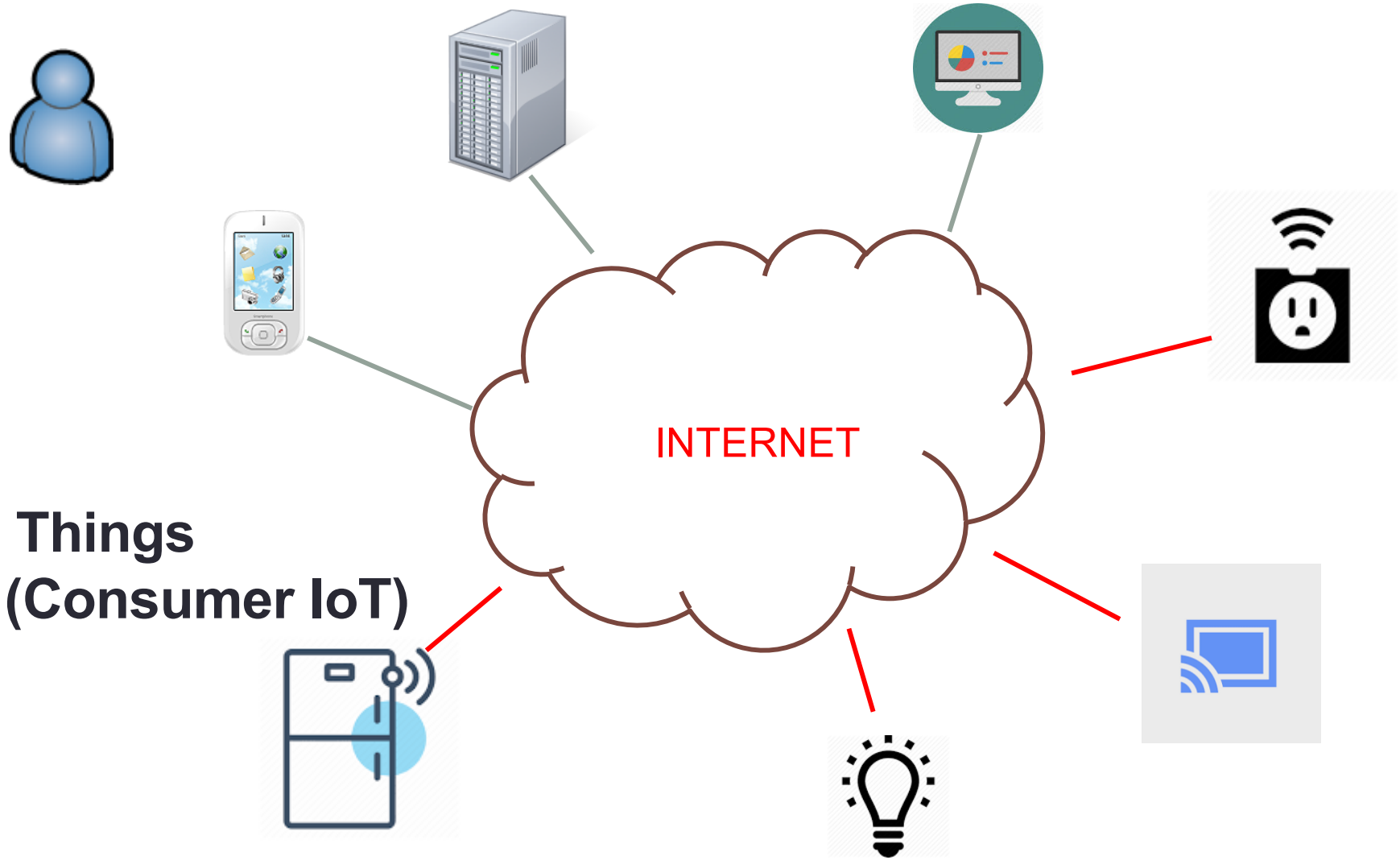
An example


Frontend
GUI
-Firebase
-GAE …
MB-as-a-Service
(push notifaction, messages, etc
-Google Maps
May be Strongly Integrated in IDE
-Wheatear Underground
-Dropbox
-Flicker
Web API, WS
(may provides SDK)
Authentication
Service
-Facebook (e.g, Oauth)
Backend

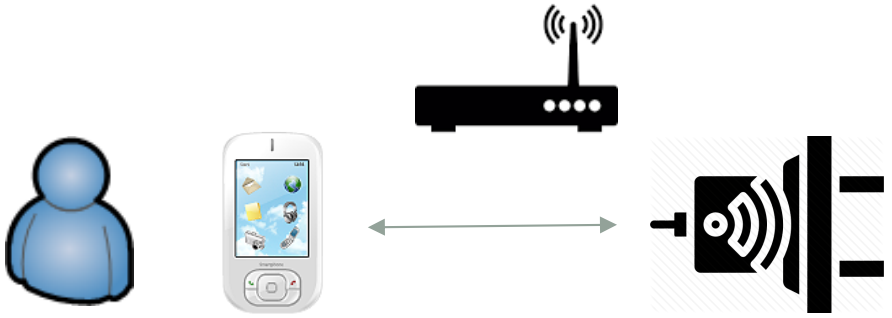
Another example



Citizens of Internet

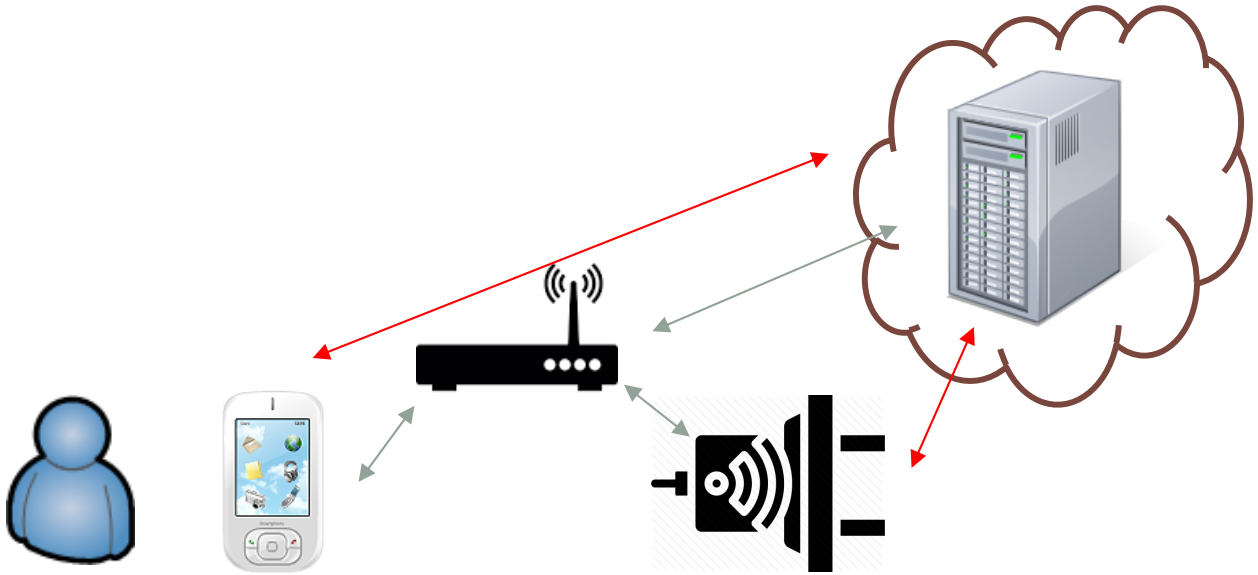


Example: enrolling a ‘smart object’

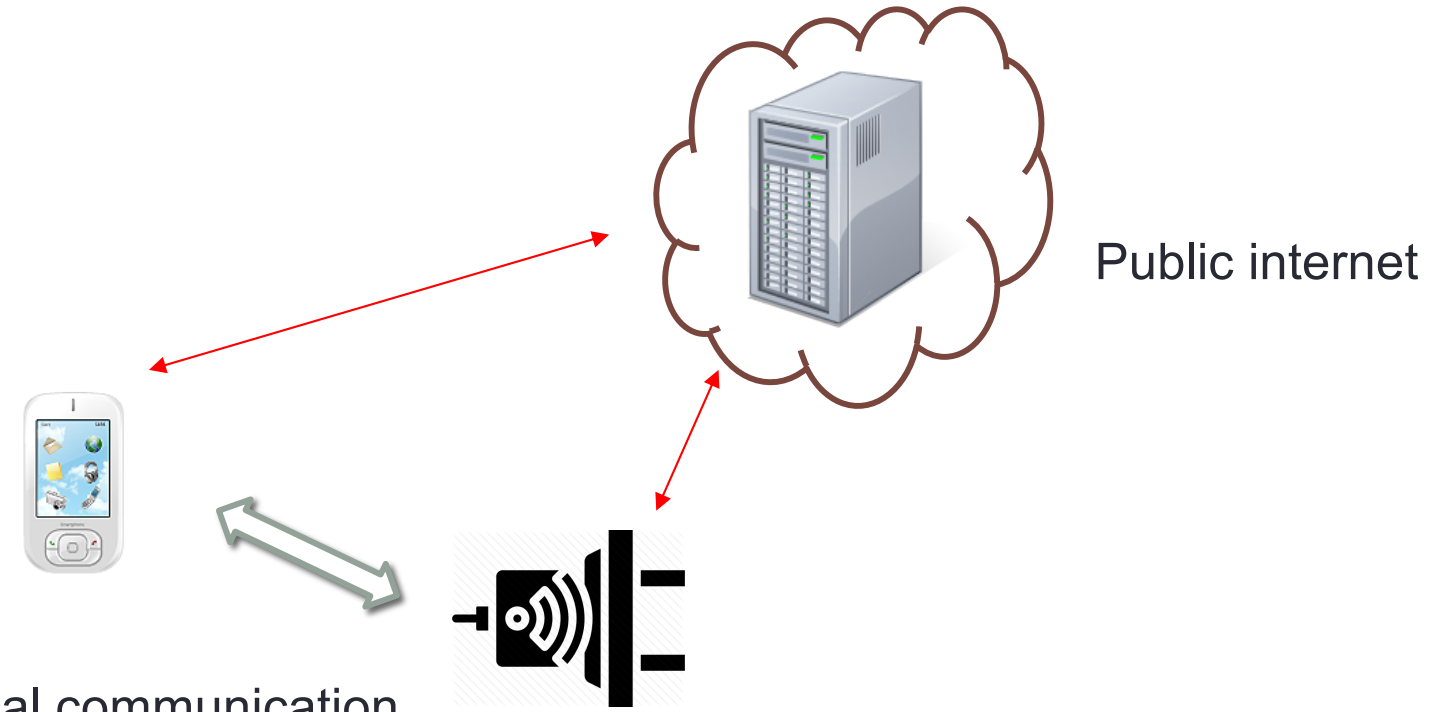


Object search for a wifi SSID

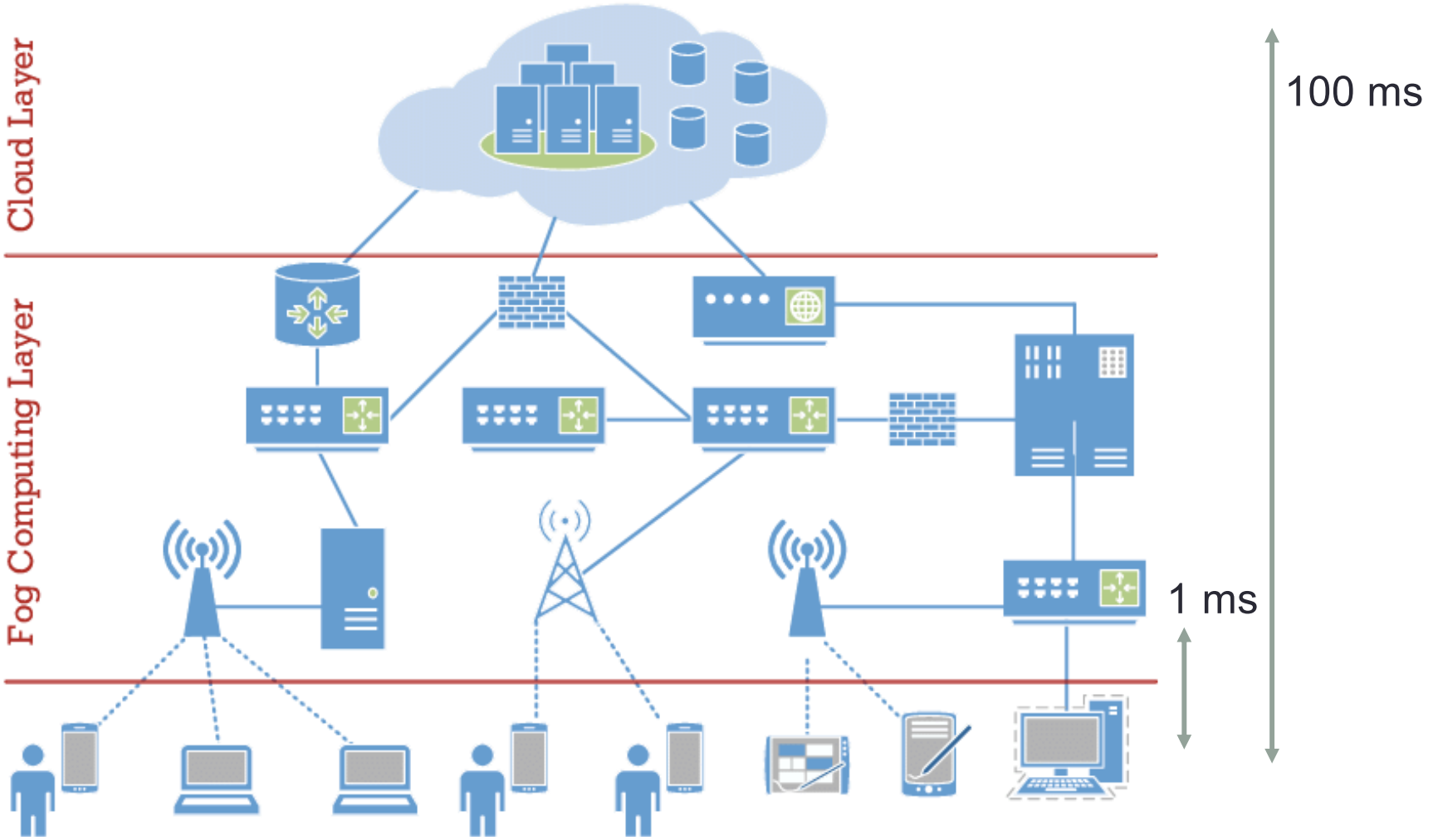
Connect to the object and configure



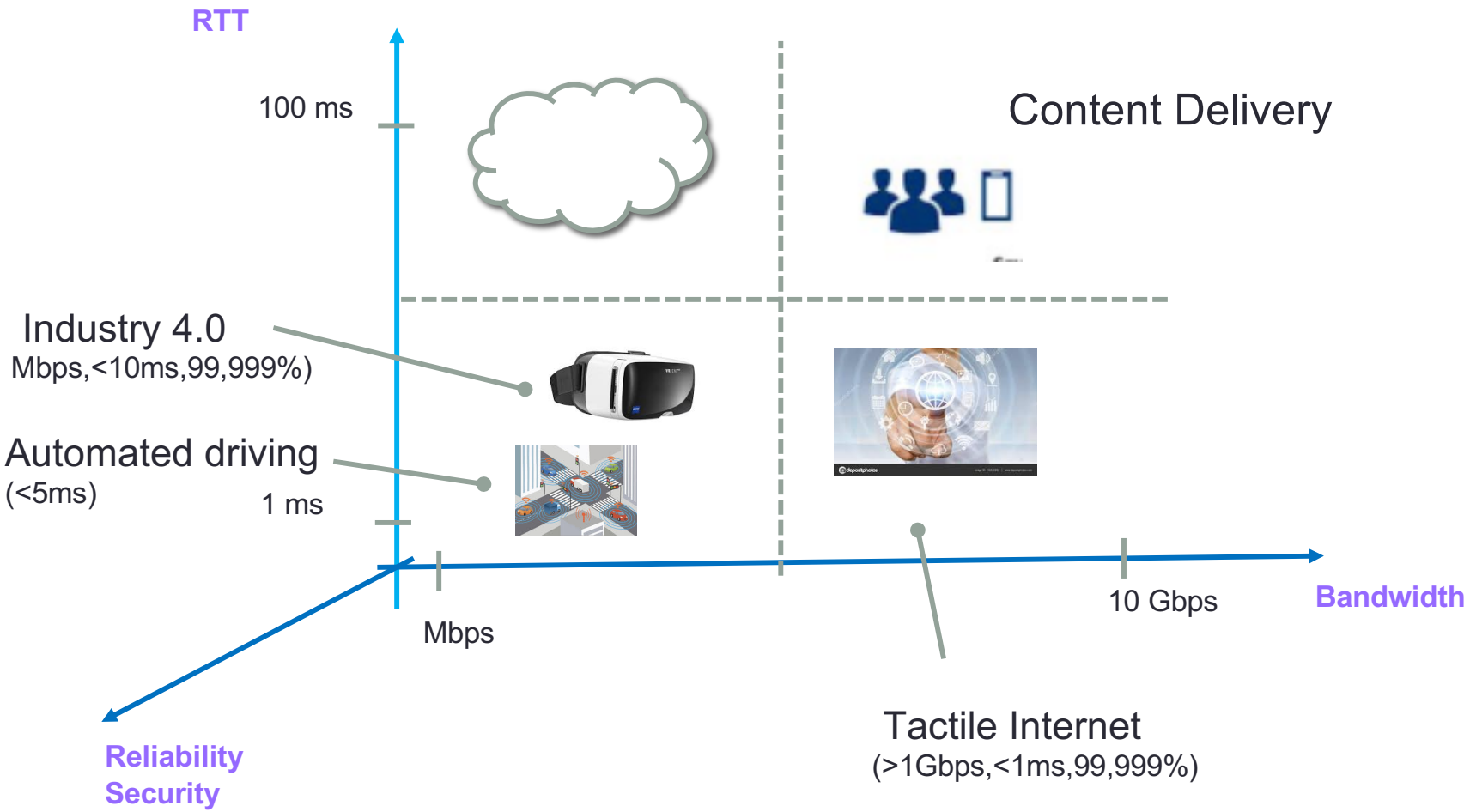
Example: enrolling a ‘smart object’



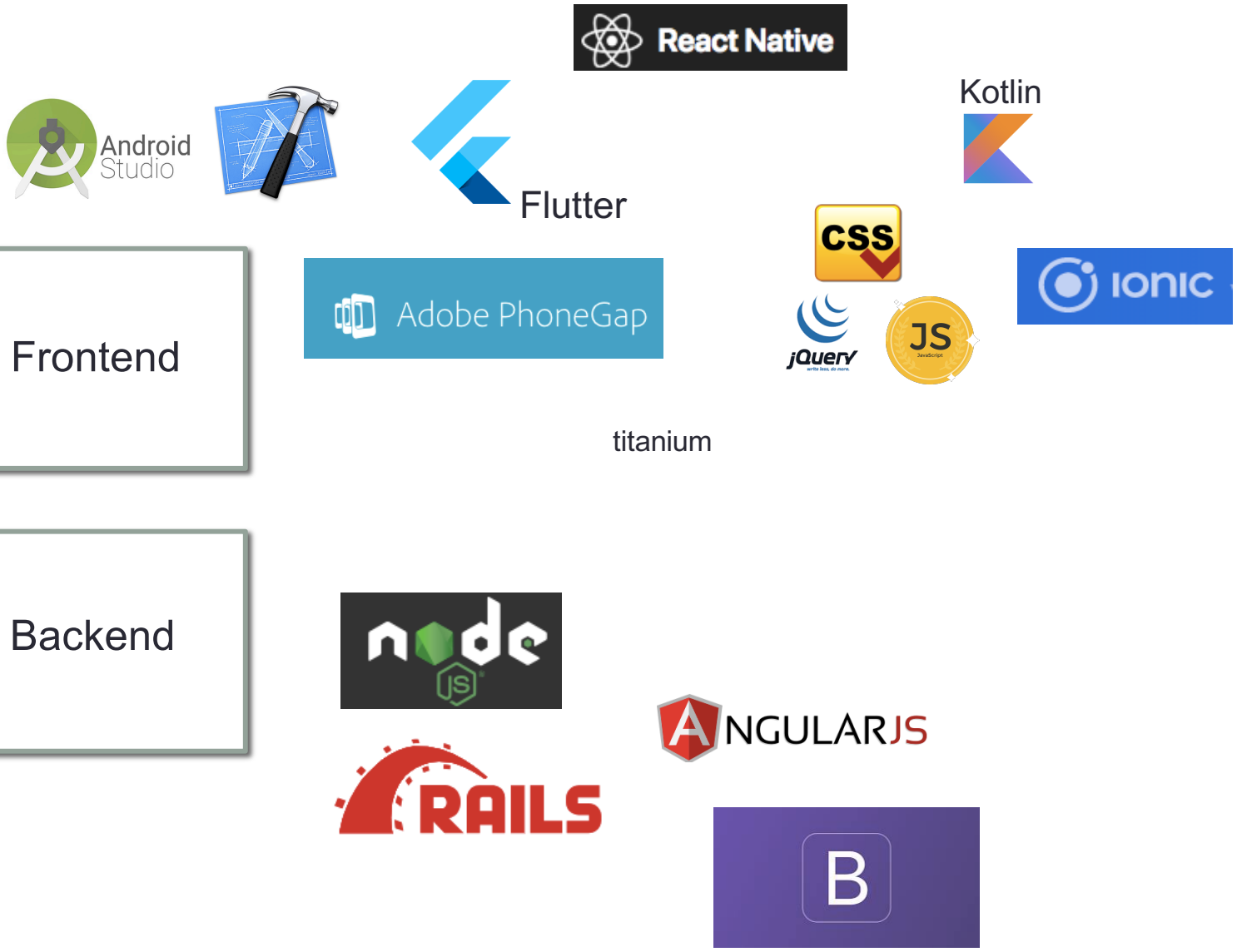
Fog Computing: moving services closer to data



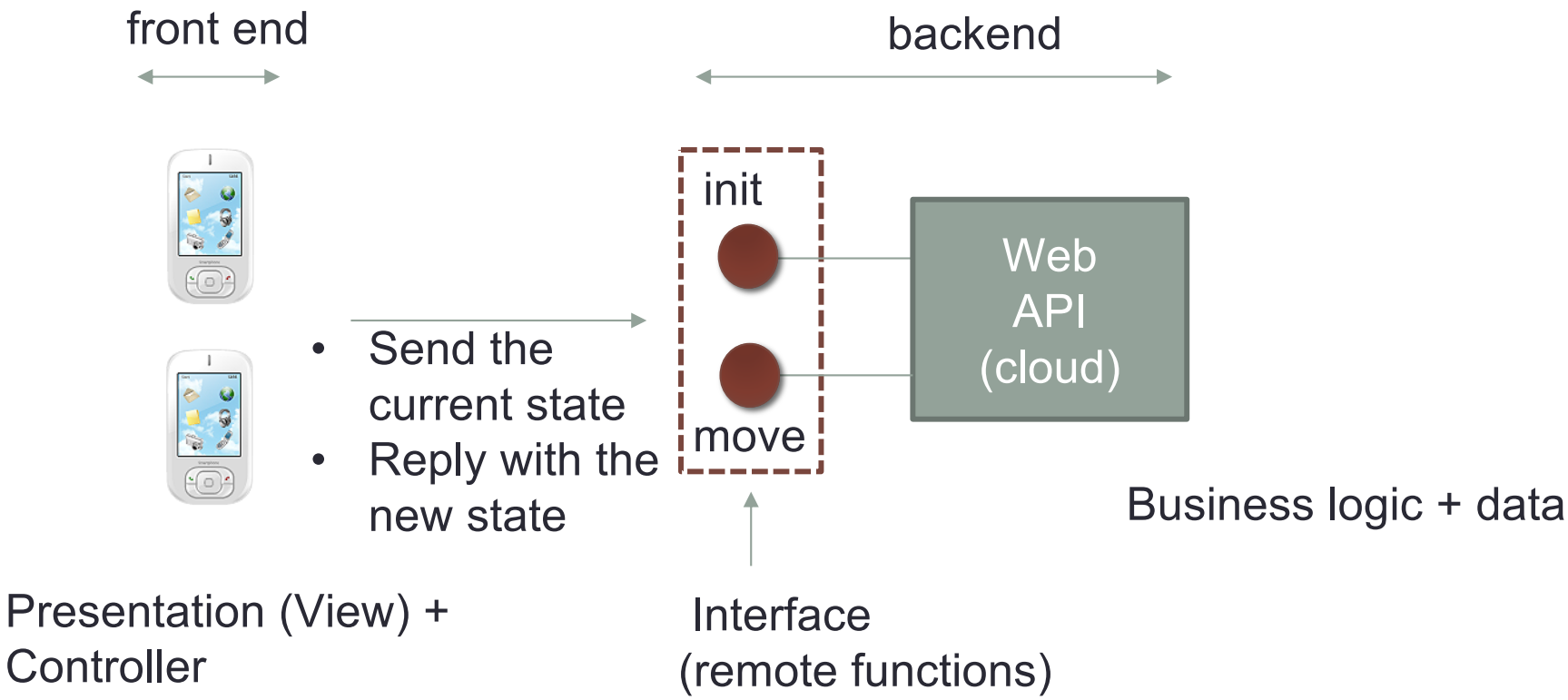
What is next?


RTT
100 ms
Content Delivery
Industry 4.0
Mbps,<10ms,99,999%)
Automated driving
(<5ms)
1 ms
Mbps
10 Gbps
Bandwidth
Reliability
Security
Tactile Internet
(>1Gbps,<1ms,99,999%)

Some popular technologies/tools



Example of an interface for a chess game



SW design patterns

- The design and implementation of an application is achieved following some **design pattern**

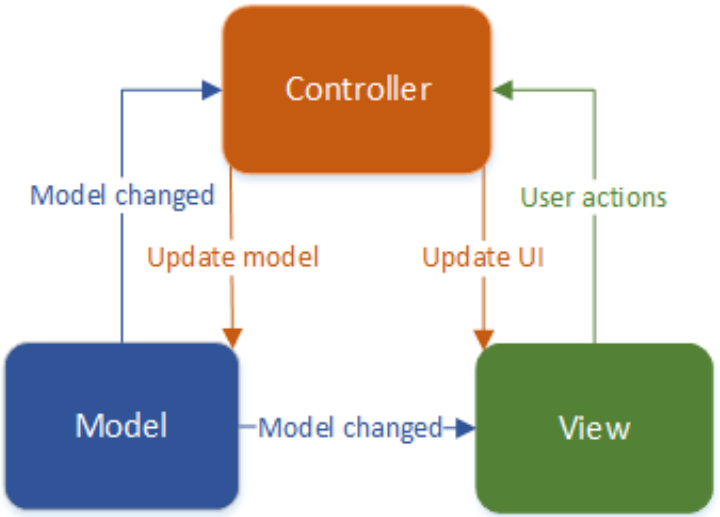
- The main goal of a pattern is to guide the software structure (architecture) in order to reach (among other goals) a high degree of **flexibility** (change or add functionality) and software **testability** (e.g. unit test)

- To this end, a patterns aims at organizing software in components that are loosely coupled (independent), so that the change in one component doesn't affect other components

Model View Controller (MVC)

	Model	View	Controller
Presentation		X	
Business		X	X
Data	X		

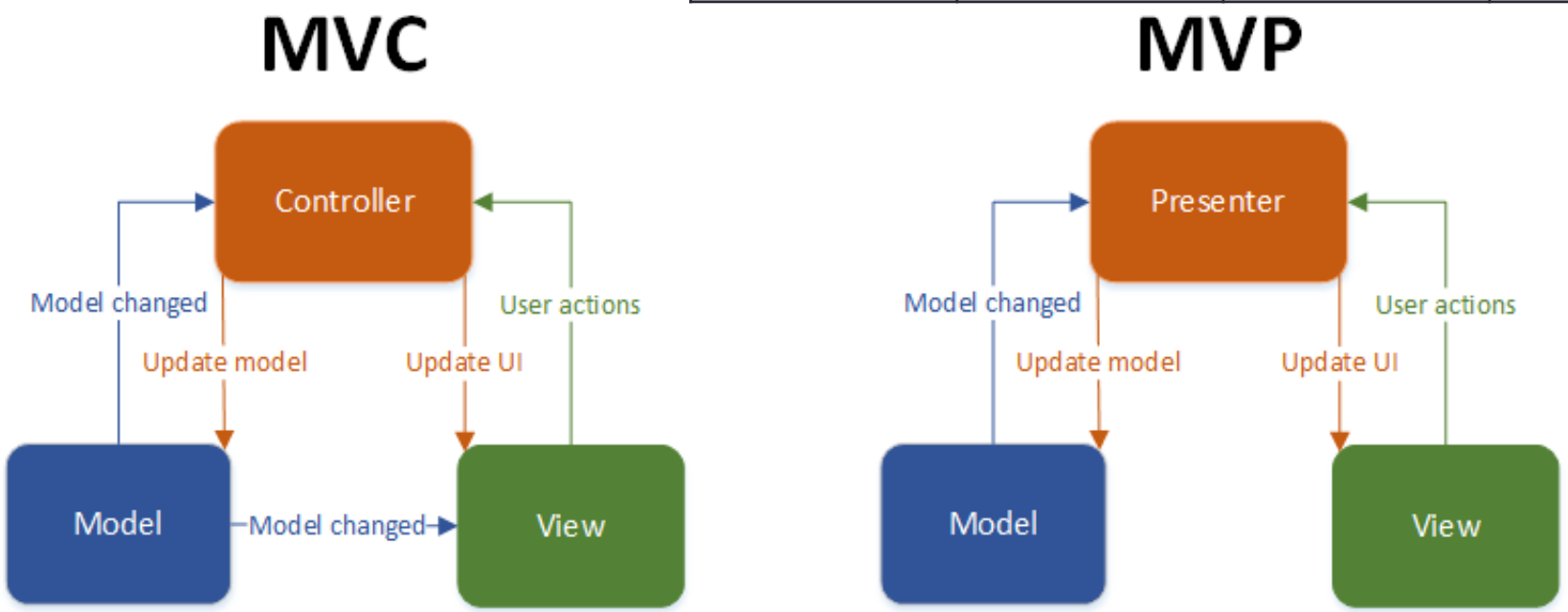
MVC



- Business logic is not confined in one place
- Model and view are coupled

Patterns: MVC and MVP

	Model	View	Controller
Presentation		X	
Business			X
Data	X		

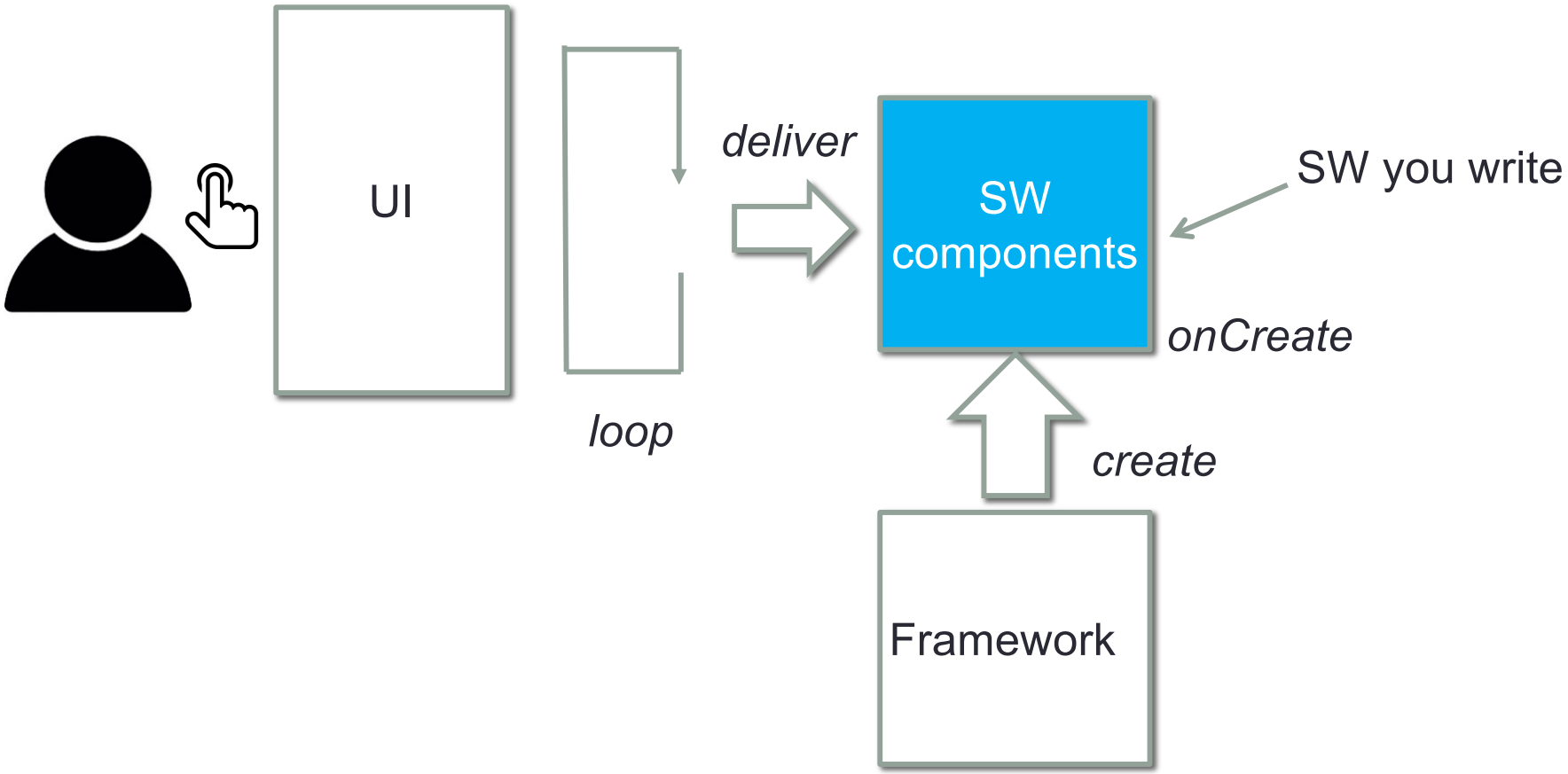


MV**P** (Presenter):
Model and View cannot communicate
Improves testability and flexibility (change/add code)
[MVC maybe fine for simple apps]

The role of framework

- A software ‘framework’ surronds the application software

An example


UI
deliver
loop
SW components
SW you write
onCreate
create
Framework

Android framework

Mobile app classification

- Web-site for mobile devices:
  - deliver web content to mobile devices, using web site style navigation
- Mobile Web:
  - web application that mimics native apps look and feel as well as navigation
- [Andorid|iOS] Native:
  - Applications installed on the devices, purched from app stores
- Hybrid:
  - Mix web and native applications
- Native:
  - C/C++

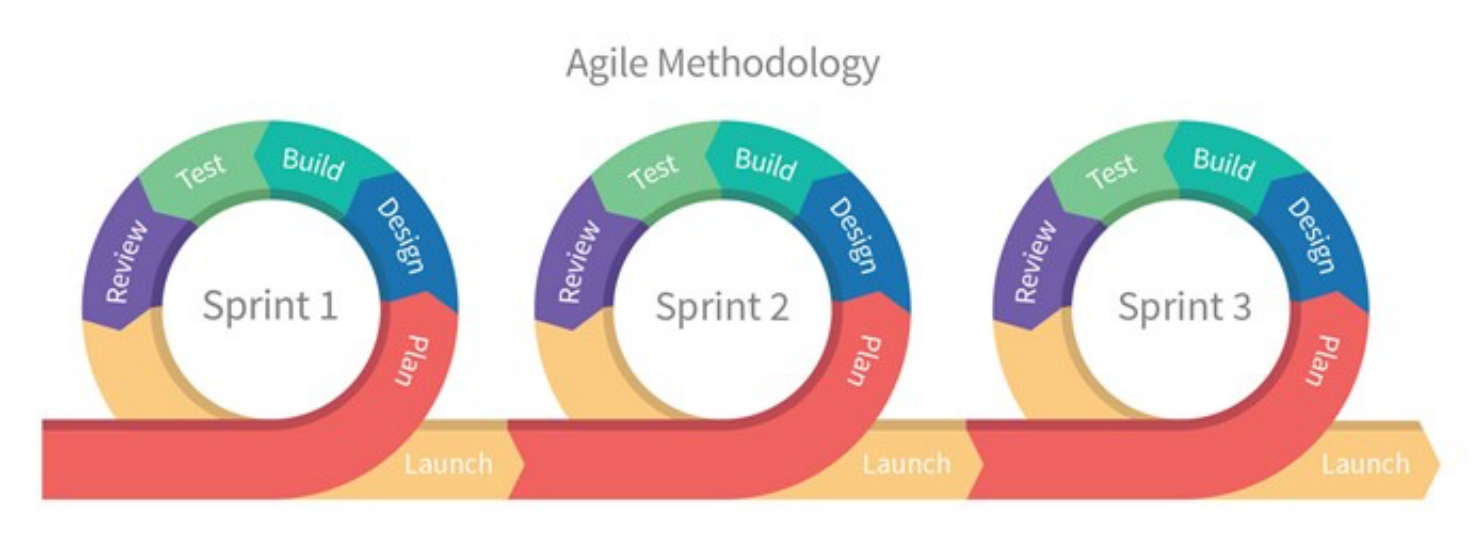
Why so many options, how to decide?

- Complexity of the app
- Productivity
- Cross-platform
- Quality of Experience
- Testability
- Performance
- Accessible features

See: https://www.mrc-productivity.com/blog/2016/06/the-mobile-app-comparison-chart-hybrid-vs-native-vs-mobile-web/

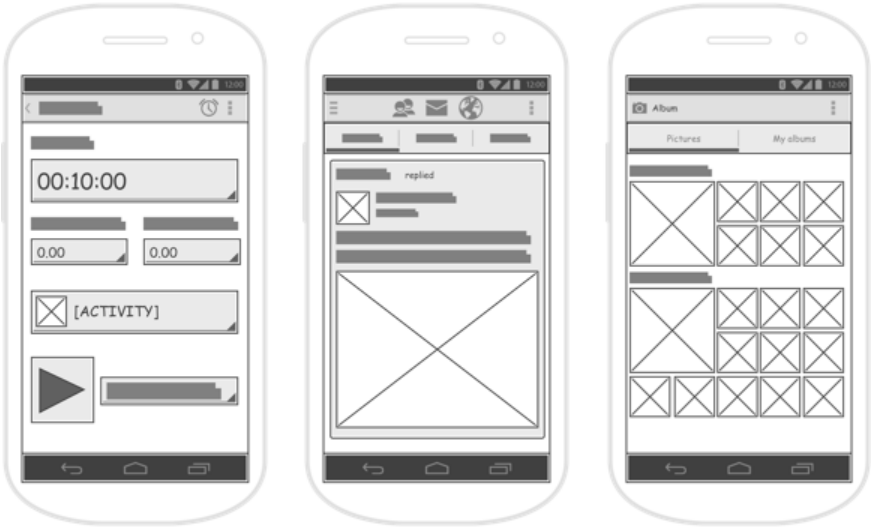
How to ‘create’ a mobile app?

- To develop a mobile app, one can use methodologies from software engineering
- The ‘agile’ methodology is one possible well established method
  - Many agile frameworks exit, e.g., Scrum
- The entire framework is studied in SE



Behaviour Driven Design, aka BDD

- A well established starting point is to define a **storyboard**
- This allows to elicit functional requirements (through **user stories**)
- **Wireframe** can be used



User stories

- User stories are a set of short descriptions about **what** the a user can do with the application
  - An application is composed of many US

- A single user story is a narrative description of a single feature.

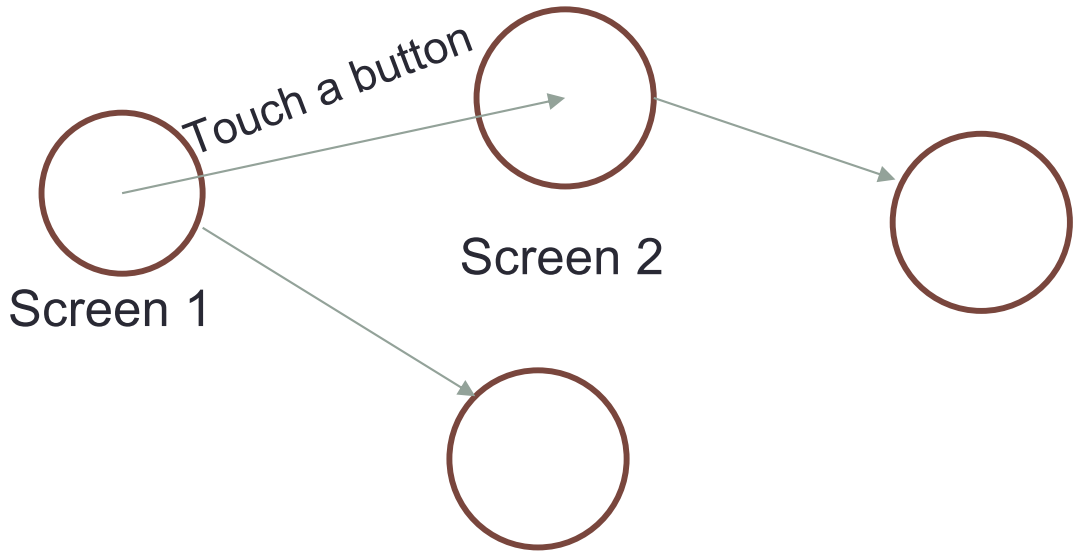
- As a user I want to login to the app so that I can use the full applications

User stories and screen

- A user story should be small enough to be represented in a single screen

Storyboard → Navigation flow

- Set of screens and their relationship



A complete example

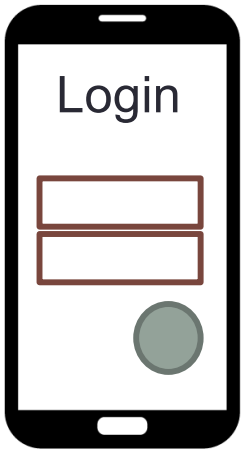
- Goal of the application: share the parking place of a family car, so that each user can know the position of the car

User stories (example)

- US1: I want to login in the app, so that only authorized users can use the app
- US2: As an authenticated user, I want to store the position of the car on a map, so that another user can later locate the car on the map
- US3: As an authenticated user, I want to know the position of the current car, so that I can pick it
- US4: As an authenticated user, I want to set the state of the car as busy or available, so that other users can know its current state

Mapping US1

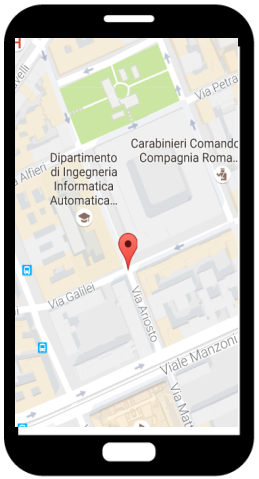
- US1: I want to login in the app, so that only authorized users can use the app



Mapping US2

- US2: As an authenticated user, I want to store the position of the car on a map, so that another user can later locate the car on the map

Mapping US3

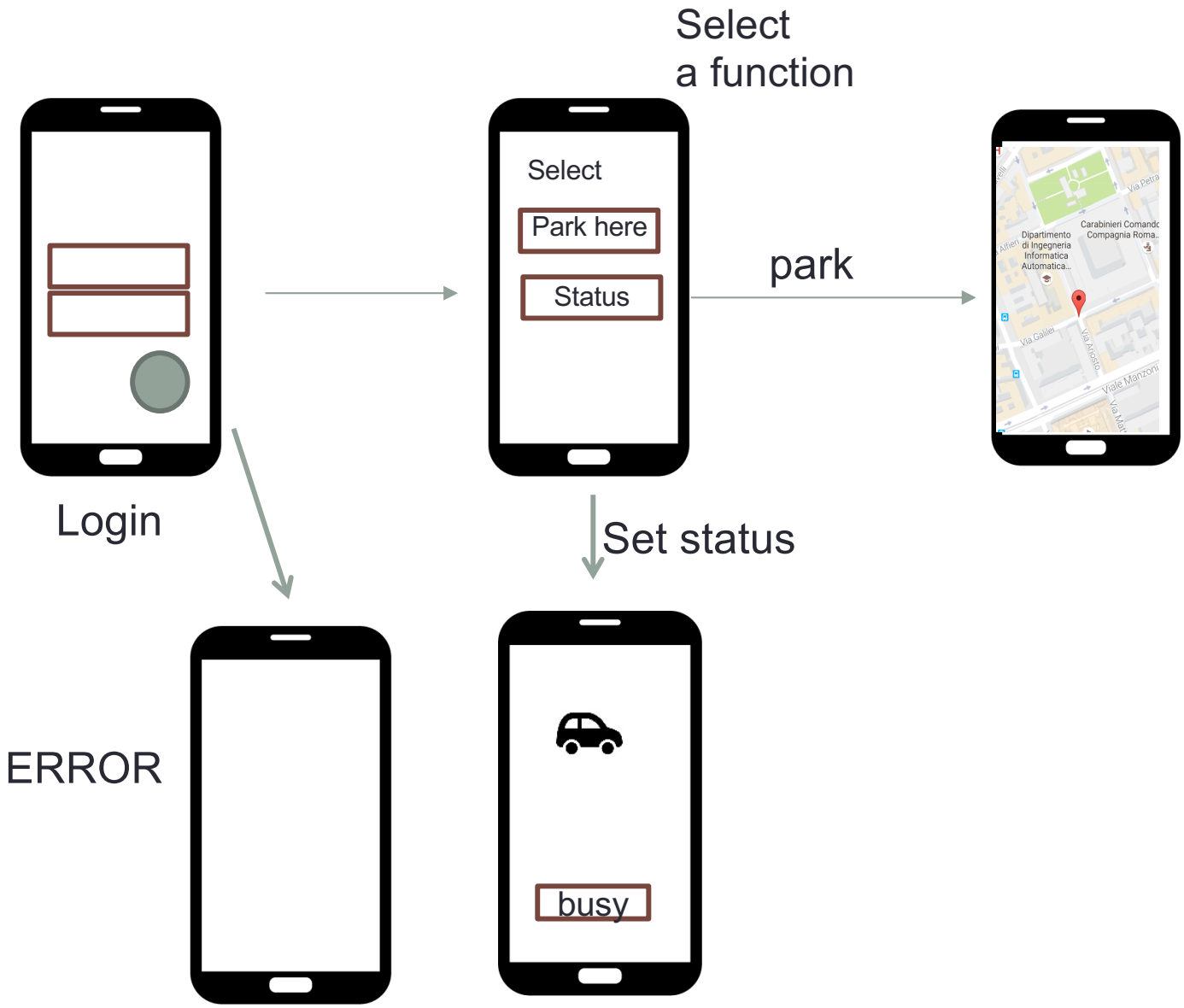


- US3: As an authenticated user, I want to know the position of the current car, so that I can pick it

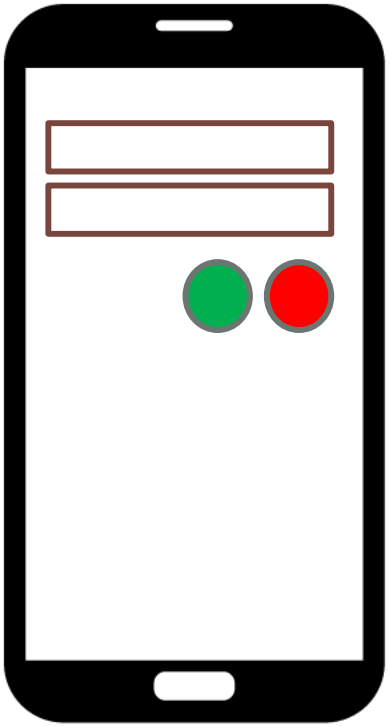
Mapping US4

- US4: As an authenticated user, I want to set the state of the car as busy or available, so that other users can know its current state

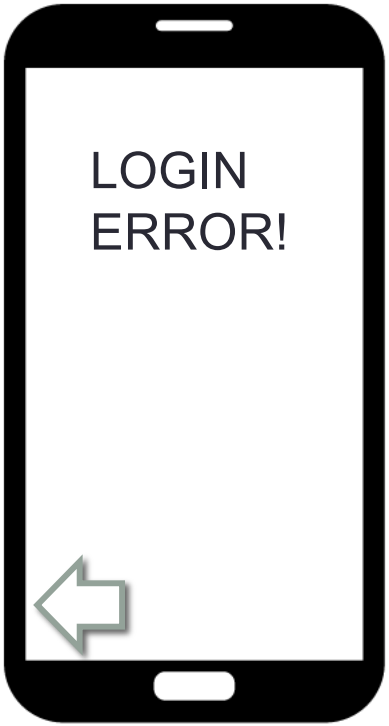
Storyboard


Select
a function
Select
Park here
Status
park
Login
Set status
ERROR
busy

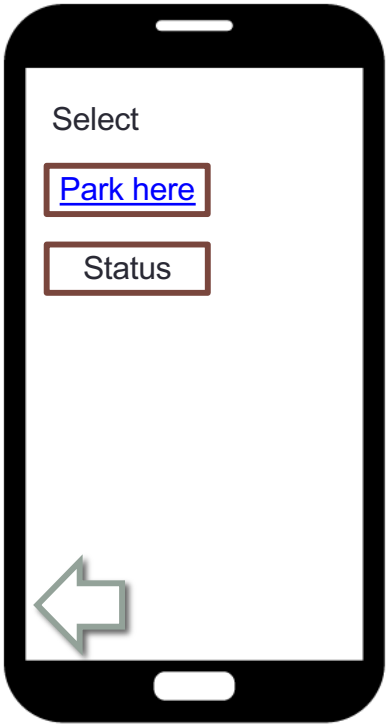
Storyboard



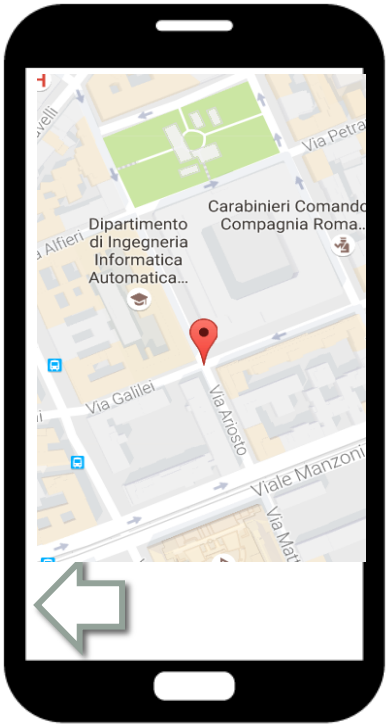
Storyboard



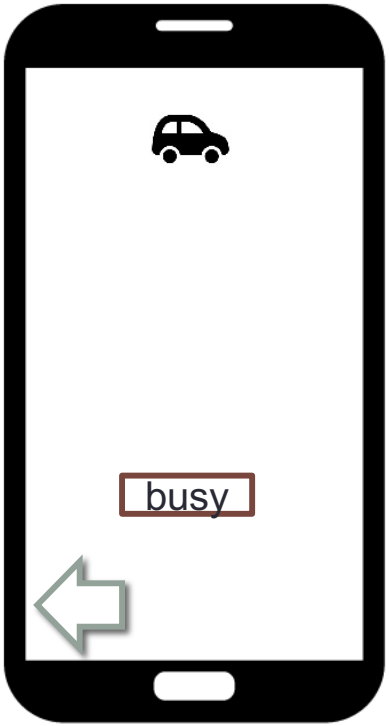
Storyboard



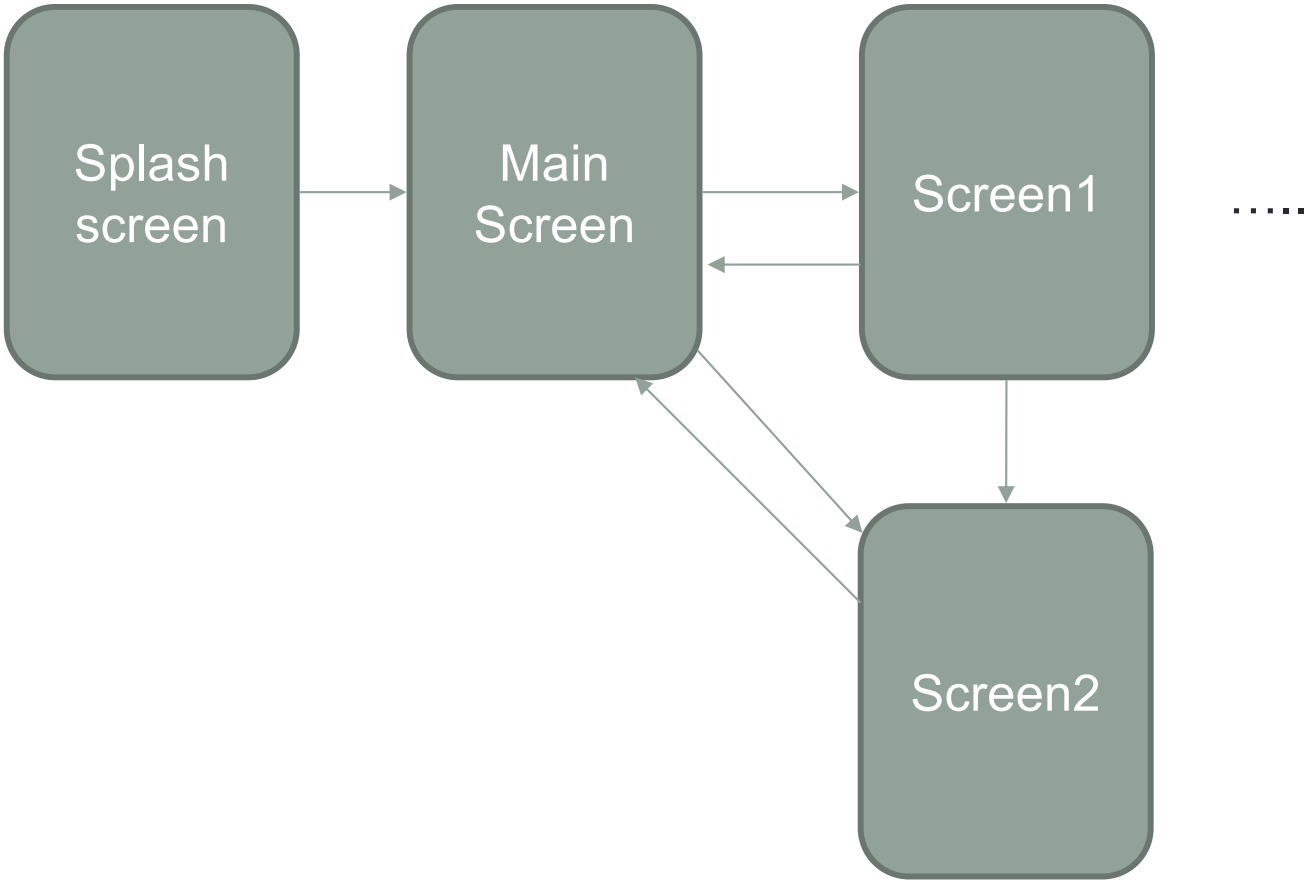
Storyboard


Dipartimento
di Ingegneria
Informatica
Automatica...
Carabinieri Comando
Compagnia Roma..
Via Galilei
Via Ariosto
Viale Manzoni

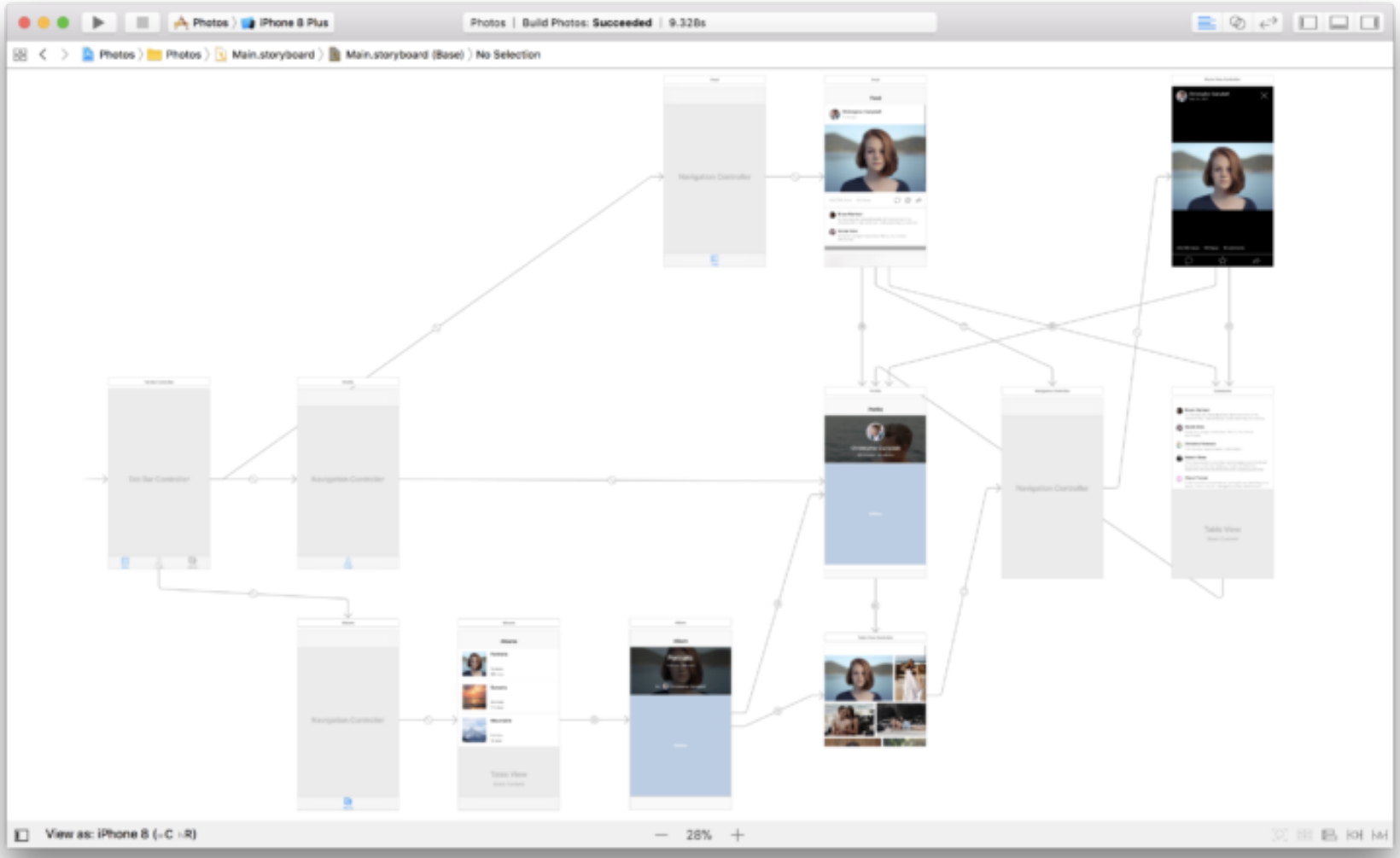
Storyboard



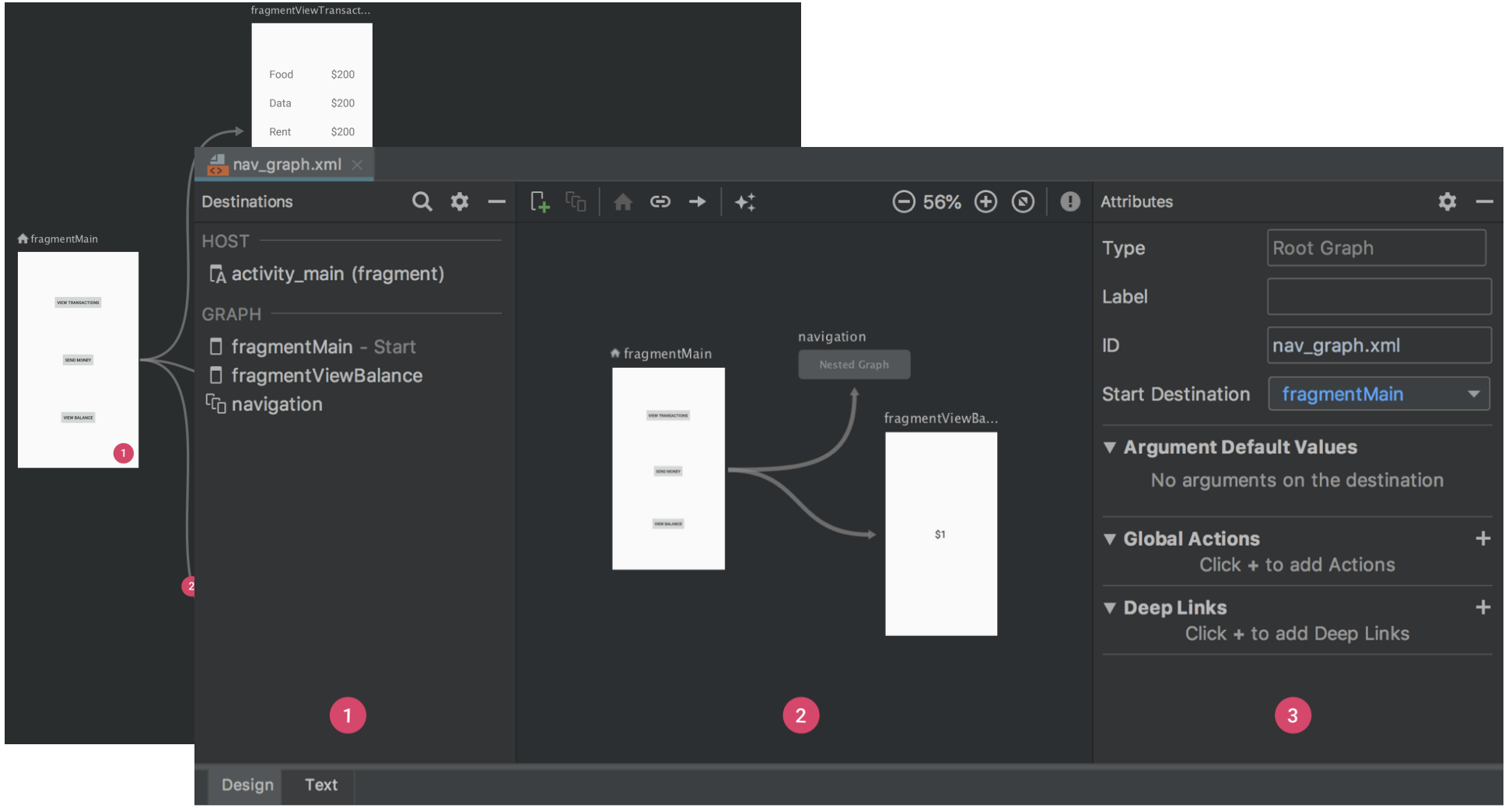
User navigation flow



Example of storyboard for iOS



Example of storyboard in Android



The storyboard facility was recently added to Android Studio and it is part of the *Jetpack* library

QUESTIONS?