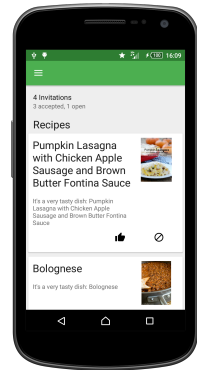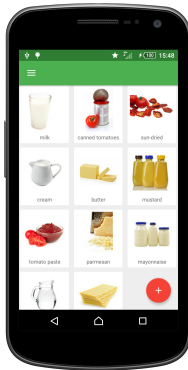# FOODSHIP GROUP UPDATE

FoodShip, a foodsharing App

Sönke Huster & Hannes Hilbert

Dresden, 16. Dezember 2016
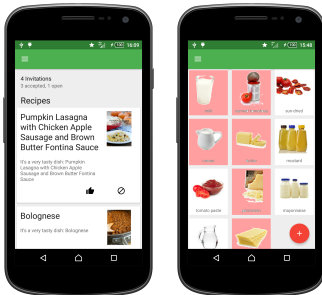
Foodship App Scenario

- App proposes having dinner with users nearby and a recipe based on the groups fridge content

Ingredient adaptation

- Suggest recipes that match many ingredients
- Example: Six of eleven ingredients needed for the best match
- Server regulary gets matching recipes


```python
def suggest_recipes(products, number=5):
    recipe_ids = []
    req = requests.get(
        'https://spoonacular-recipe-food-nutrition-v1.p.mashape.com/recipes/findByIngredients'
        '?fillIngredients=false'
        '&ingredients={}'
        '&limitLicense=false'
        '&number={}'
        '&ranking=1'.format(products, number),
        headers=headers)
    if req.status_code == requests.codes.ok:
        recipes = req.json()
        for recipe in recipes:
            recipe_ids.append(get_recipe_detail(recipe['id']))
    else:
        logging.error("Request failed!")
        logging.error(req.request)
    return recipe_ids
```

Location adaptation

- Find groups of people nearby
- Server calculates groups by location
- Technology used: PostgreSQL Database with PostGIS Extension for location features
  - PostGIS function example used in SQL Query: ST_DISTANCE(user1.location, user2.location)<1500
  - Returns TRUE if user1 is in a 1.5km range of user2
- Example: user_id has X possible group_members in a range of max_distance

	user_id	group_member	max_distance
1	3	3	1365.07881417
2	4	3	1059.40901467
3	2	2	1059.40901467
4	5	2	1365.07881417

Adaptation of Communication

- "Adapt the way data is exchanged between distributed components"
- A ConnectivityManager checks if a NetworkConnection is availiable:
  - If there is one the Call gets executed
  - On Error or with no Connection the Calls are persisted
- We use com.birbit.android.jobqueue for queueing API Calls

```
SetUserLocationJob setUserLocationJob = new SetUserLocationJob(Utils.getUserId(context), latLng.latitude, latLng.longitude);
Params params = new Params( priority: 0)
        .setRequiresNetwork(true)
        .setPersistent(true);
setUserLocationJob.setParams(params);
FoodshipJobManager.getInstance(context).addJobInBackground(setUserLocationJob);
```

Adaptation of Connectivity

- Push notification triggered by our server
- App then prefetches group information and recipe pictures
- Data is persisted in internal storage and in cache for better user experience

```java
package de.foodshippers.foodship;

import ...

/**...*/
public class MyFirebaseMessagingService extends FirebaseMessagingService {

    private static final String TAG = MyFirebaseMessagingService.class.getSimpleName();

    @Override
    public void onMessageReceived(RemoteMessage firebaseMassage) {
        super.onMessageReceived(firebaseMassage);
        //Send Notification to User
        sendNotificationtoUser();
        //Gets the dataController
        GroupDataController dataController = GroupDataController.getInstance(getApplicationContext());
        //Sets new GroupID
        dataController.setGroupId(Integer.decode(firebaseMassage.getData().get("group_id")));
        //Starts Prefetching of GroupData
        dataController.prefetch();
    }

    private void sendNotificationtoUser() {...}

}
```
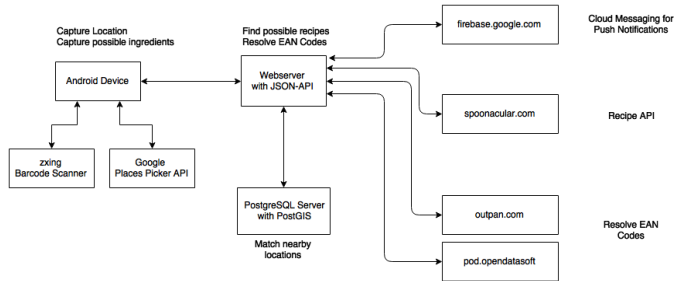
Architecture & Technologies

Whats next?

- Better integration of the dinner groups into the app
- Image size adaptation depending on the network speed
- Testing with more devices in real environment
- Final Presentation