Müllner, Huber, Kralicek, Deubler

Cross-platform

Bluetooth

- Tried various Libraries
  - No Cross-Platform, which we consider mandatory
  - Mostly BLE for Heart rate monitors, etc.
  - Buggy or outdated
  - Couldn't even get Android to Android working

- cordova-plugin-chrome-apps-bluetoothSocket
- cordova-plugin-networking-bluetooth
- cordova-plugin-ble-central
- BluetoothSerial

Network

- Platform independent
- Global matchmaking
- Everybody is de facto online
- Common use case
- Well documented and many possibilities
- We need a backend anyway… (more later)

Network

- Use of socket.io
  - enables real-time bidirectional event-based communication
  - up-to-date and well maintained
  - easy to use with NodeJS Server
  - easy to use with Cordova
  - NPM package

socket.io

Network

- socket.io server runs as Node-app inside a Docker container
- nginx container in front as http proxy
  - forwards traffic to container
  - accessible through http://kdeubler.at/socket.io/
- a little bit of work to get websockets through

Network

- app.js starts the server
- include socket.io
- include game.js
  - contains listeners
  - execute „network logic“
- defined in package.json
- listens on incoming connections

```
var server = require('http').createServer();
var io = require('socket.io')(server);

var battleship = require('./game');

io.sockets.on('connection', function (socket) {
    console.log('Client connected');

    battleship.initGame(io, socket);
});

server.listen(8082);
```

Network

- define functions that can be called by the client

```
gameSocket.on('hostCreateNewGame', hostCreateNewGame);
gameSocket.on('startGame', startGame);
gameSocket.on('playerJoinGame', playerJoinGame);
gameSocket.on('playerPlaceShipsFinished', playerPlaceShipsFinished);

gameSocket.on('playerFire', playerFire);
gameSocket.on('playerFireResult', playerFireResult);
gameSocket.on('playerFieldSelection', playerFieldSelection);
gameSocket.on('playerGameEnded', playerGameEnded)
```

- socket.io manages sockets in „rooms“
  - perfect for our case
  - host creates room
  - second player joins room
  - emit messages within this room
  - different broadcasting methods
    - send to all except sender

```
function startGame(gameId) {
    console.log("All Players Present. Starting game...");
    var sock = this;
    var data = {
        mySocketId : sock.id,
        gameId : gameId,
    };

    io.sockets.in(data.gameId).emit('beginGame', data);
};
```

Network

```
socket = io('http://localhost:8082');
socket.on('gameHosted', gameHosted);
socket.on('beginGame', beginGame);
socket.on('onError', onError);

socket.on('otherPlayerFired', otherPlayerFired);
socket.on('otherPlayerGameEnded', otherPlayerGameEnded);
socket.on('otherPlayerFireResultReceived', otherPlayerFireResultReceived);
socket.on('otherPlayerFieldSelected', otherPlayerFieldSelected);
```

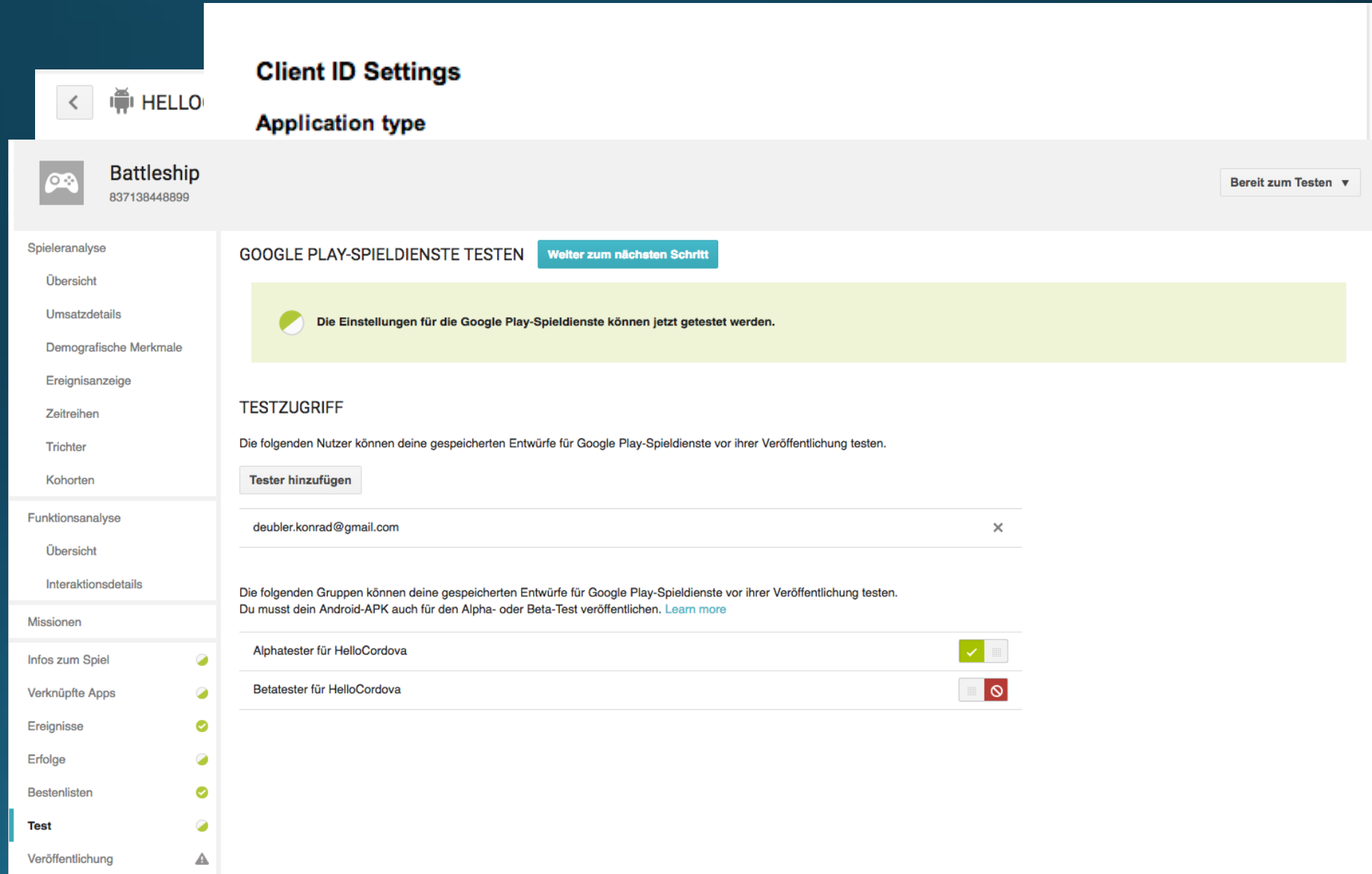
```
function joinGame() {
    console.log("Join game");

    var gameData = {
        gameId: document.getElementById("myConsole").value,
        playerName: "Knusbert"
    }

    socket.emit('playerJoinGame', gameData);
}
```

```
function playerJoinGame(data) {
    console.log('Player ' + data.playerName + ' attempting to join game: ' + data.gameId );

    // A reference to the player's Socket.IO socket object
    var sock = this;

    try {
        // Look up the room ID in the Socket.IO manager object.
        var room = io.sockets.adapter.rooms[data.gameId];

        if (room != undefined){
            // attach the socket id to the data object.
            data.mySocketId = sock.id;

            // Join the room
            sock.join(data.gameId);

            console.log('Player ' + data.playerName + ' joined game: ' + data.gameId );

            // Directly start game
            startGame(data.gameId);
```

Google Play Services
iOS Game Center

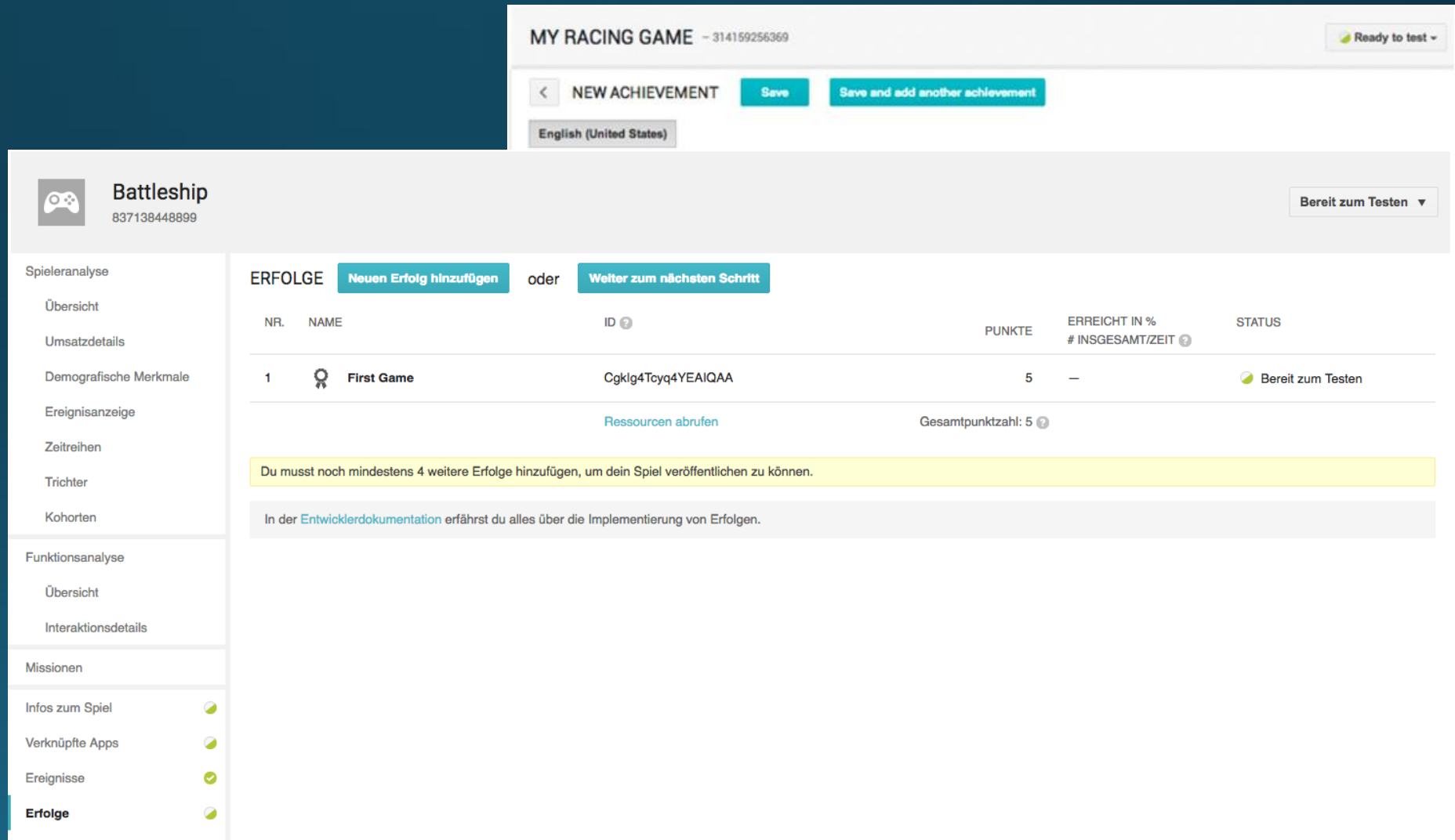
Google Developer Console

- Create
- Link
- Authorize
- Test



Google Developer Console II

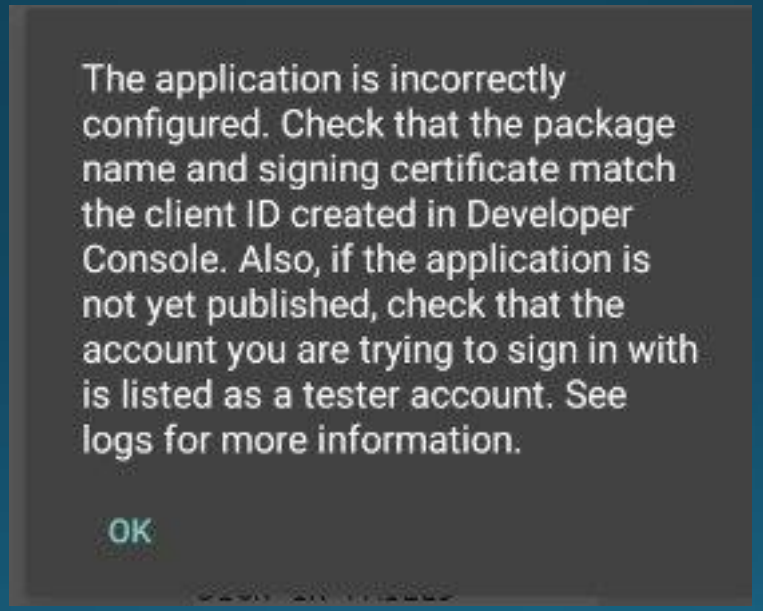
- Add



Plugins

- artberri/cordova-plugin-play-games-services
- cranberrygame/cordova-plugin-game
- ludei/atomic-plugins-googleplaygames
- cocoon-google-play-games-social-plugin
- floatinghotpot/google-play-services

Result


The application is incorrectly configured. Check that the package name and signing certificate match the client ID created in Developer Console. Also, if the application is not yet published, check that the account you are trying to sign in with is listed as a tester account. See logs for more information.
OK

Alternatives

- Facebook Games API
  - Plugins available
- Write new Plugin for Google Play Services
- Own Webservice

Pitfalls / things I liked

- Plugins are written for one app and not maintained
- Plugins often copies of other plugins with few “enhancement”
- Google Play Service Documentation

Good starting points

- Google Play Services API Documentation
- iOS Developer Library

Demo

Questions?