⚠ This repository has been removed and cannot be used anymore.

Bezirk-Bosch/Middleware (https://github.com/Bezirk-Bosch/Middleware)

developer-friendly, user-centric, and privacy-respecting glue for the Internet of Things

http://www.bezirk.com/ (http://www.bezirk.com/)

License: MIT (/licenses/MIT)

Language: Java (/languages/Java)

Bezirk

The Bezirk middleware is developer-friendly, user-centric, and privacy-respecting glue for the Internet of Things.

Build Dependencies

- Latest version of the JDK
- (Optional) Latest version of Android Studio

To build and test the middleware run: `./gradlew check`

To build the Bezirk API JavaDocs: `./gradlew :core:bezirk-middleware-api:javadoc`

To build and run the Java SE test app (convenient for quick sanity checks): `./gradlew :java:testApp:run`

Vision

The more compelling aspects of the Internet of Things dream involve cooperation between devices. With this cooperation, we can achieve dreams more complex than are currently possible. Cooperation can enable a world where we offload our mundane tasks to our Things, with outcomes customized to our needs and desires. Cooperation can also optimize business and industrial processes using accurate and timely data. These dreams need an ecosystem where devices seamlessly interoperate and are easy for the right (and only the right) entities to access.

In practice, devices are diverse in their manufacturers, functions, and use cases, and they often have buggy, poorly documented APIs that do not work together. How do we fulfill the IoT dream if the basic building blocks are clumsy and difficult to work with?

The Bezirk ecosystem aims to solve these problems. The middleware in this repository forms its heart. The Bezirk middleware implements cloudless and brokerless publish-and-subscribe in Java. It also includes basic security and identity management building blocks. Together, these components enable secure, seamless interoperability. The ecosystem includes the following components to bootstrap interoperability:

- *Zirks* use the middleware to communicate with hardware or other computing devices within the Bezirk ecosystem
- *Events* are the unit of communication between Zirks
  - Adapter Zirks (https://github.com/Bezirk-Bosch/AdapterZirks) are wrappers that enable existing hardware (e.g., Philips Hue lights) to be controlled using Bezirk events.
  - Hardware events (https://github.com/Bezirk-Bosch/HardwareEvents) provide a uniform interface for accessing IoT devices, including beacons, environmental sensors, lights, and others.

Example

With the middleware, events, and adapter Zirks, very little code is required to write a Zirk that detects and actuates all Lightify lights on a network:

Project Statistics

Sourcerank	**6 (/github/Bezirk-Bosch/Middleware/sourcerank)**
Repository Size	**13.7 MB**
Stars	**4 (https://github.com/Bezirk-Bosch/Middleware/stargazers)**
Forks	**0 (https://github.com/Bezirk-Bosch/Middleware/network)**
Watchers	**15 (https://github.com/Bezirk-Bosch/Middleware/watchers)**
Open issues	**15 (https://github.com/Bezirk-Bosch/Middleware/issues)**
Dependencies	**29**
Contributors	**6 (/github/Bezirk-Bosch/Middleware/contributors)**
Tags	**19 (/github/Bezirk-Bosch/Middleware/tags)**
Created	**Apr 8, 2016**
Last updated	**May 7, 2018**
Last pushed	**Mar 10, 2017**

Top Contributors See all (/github/Bezirk-Bosch/Middleware/contributors)


(/github/rishabhgulati) (/github/vimalanandan
(/github/punithmailme)
(/github/maa
bez
(/github/nasrinivasan) (/github/kevinhhbos

Recent Tags See all (/github/Bezirk-Bosch/Middleware/tags)

v3.2.1-snapshot (https://github.com/Bezirk-	March 10

```
final Bezirk bezirk = BezirkMiddleware.registerZirk("Lightify Zirk");

final EventSet lightEvents = new EventSet(LightsDetectedEvent.class);

lightEvents.setEventReceiver(new EventSet.EventReceiver() {
    @Override
    public void receiveEvent(Event event, ZirkEndPoint sender) {
        if (event instanceof LightsDetectedEvent) {
            for (final Light light : ((LightsDetectedEvent) event).getLights()) {
                bezirk.sendEvent(new TurnLightOnEvent(light));

                new Timer().schedule(new TimerTask() {
                    @Override
                    public void run() {
                        bezirk.sendEvent(new TurnLightOffEvent(light));
                    }
                }, 2000);
            }
        }
    }
});

bezirk.subscribe(lightEvents);

Set<String> gateways = LightifyAdapter.discoverGateways();
try {
    new LightifyAdapter(bezirk, gateways.toArray(new String[gateways.size()])[0]);
} catch (IOException e) {
    logger.error("Failed to connect to lightify gateway", e);
}
```

Using Philips Hue products instead? Simply switch the last two (non-exception related) lines of code to use the Hue adapter. This example even grows in capability without code modifications. If another Zirk on the same network were to implement support for yet another model of lights and broadcasts `LightsDetectedEvent`, our example Zirk will seamlessly receive its messages and actuate the new lights.

The Bigger Picture

Many of the examples above have been smart home oriented. We are thinking bigger. What if the Zirks you use sent observations about how you use them to a personalization Zirk you own? This personalization Zirk could then build profiles based on how you interact with your world. What if instead of waiting to order at a coffee shop, the store could ask your personalization Zirk about your favorite drink as you walk in? What if your taxi knew what temperature the car should be and what to set the radio to as it pulls up to pick you up? What if the shipping processes for your medicine or groceries were smart enough to ensure unbroken cold chains (https://en.wikipedia.org/wiki/Cold_chain)? What if your city could dynamically adjust stop light timings based on current traffic? What if your doctor could tell what is going on every time your blood pressure is too high? All of this can be built on top of the Bezirk middleware.

You might be thinking to yourself: "What about privacy concerns? I'm not sure I'm comfortable having every action I take observed to build profiles." This is valid and another problem related to compelling IoT visions that we're trying to solve. The Bezirk middleware aims to be privacy-respecting in the sense that it puts the user in control of their own data, while taking care of the nitty-gritty security details automatically. We don't force users' data to go through the cloud, and users can decide the scope within which their Zirks can communicate. As a result, all of the users' information remains completely in their control. While parts of this vision are not implemented yet, we aim to make it as easy on the user as possible. We firmly believe that people should not have to possess expert-level security awareness to use the IoT safely, securely, and privately.

Helping Out

Please peruse the wiki (https://github.com/Bezirk-Bosch/Middleware/wiki) to learn more about what we are working on, review design details for implemented and prospective features, and, most importantly, learn how you can contribute to the effort.

Dependencies

android/android-libraries/android-zirk-proxy/build.gradle (https://github.com/Bezirk-Bosch/Middleware/blob/master/android/android-libraries/android-zirk-proxy/build.gradle)

Kind	Requirements	Latest Stable	Latest Release	Licenses
com.google.code.gson:gson (/maven/com.google.code.gson:gson)	runtime	2.7	2.8.6	2.8.6	Apache-2.0 (/licenses/Apache-2.0)

android/android-starter/build.gradle (https://github.com/Bezirk-

Bosch/Middleware/releases/tag/v3.2.1-snapshot)	2017
v3.2.0-snapshot (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.2.0-snapshot)	March 10 2017
v3.1.1 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.1.1)	January 19, 2017
v3.1.1-snapshot (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.1.1-snapshot)	Decembe 21, 2016
v3.1.0 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.1.0)	Decembe 09, 2016
v3.0.2 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.2)	Novembe 08, 2016
v3.0.1 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.1)	October 18, 2016
v3.0.0 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.0)	October 14, 2016
v3.0.0-alpha+025-snapshot (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.0-alpha+025-snapshot)	October 05, 2016
v3.0.0-alpha+024 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.0-alpha+024)	Septembe 28, 2016
v3.0.0-alpha+023 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/v3.0.0-alpha+023)	Septembe 21, 2016
3.0.0-alpha+021 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/3.0.0-alpha+021)	Septembe 09, 2016
3.0.0-alpha+21 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/3.0.0-alpha+21)	Septembe 09, 2016
3.0.0-alpha+019 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/3.0.0-alpha+019)	Septembe 06, 2016
3.0.0-alpha+016 (https://github.com/Bezirk-Bosch/Middleware/releases/tag/3.0.0-alpha+016)	August 19 2016

Bosch/Middleware/blob/master/android/android-starter/build.gradle)	Kind	Requirements	Latest Stable	Latest Release	Licenses
com.github.tony19:logback-a... (/maven/com.github.tony19:logback-android-classic)	runtime	1.1.1-2		1.1.1-6	EPL-1.0 (/licenses/EPL-1.0)/LGPL-2.1+ (/licenses/LGPL-2.1+)
com.j256.ormlite:ormlite-an... (/maven/com.j256.ormlite:ormlite-android)	runtime	4.48		5.0	ISC (/licenses/ISC)
com.j256.ormlite:ormlite-core (/maven/com.j256.ormlite:ormlite-core)	runtime	4.48		5.0	ISC (/licenses/ISC)
org.slf4j:slf4j-api (/maven/org.slf4j:slf4j-api)	runtime	1.7.12	1.7.28	2.0.0-alpha0	MIT (/licenses/MIT) ⚠

android/android-test-apps/bezirk-android-testapp/build.gradle (https://github.com/Bezirk-Bosch/Middleware/blob/master/android/android-test-apps/bezirk-android-testapp/build.gradle)

Kind	Requirements	Latest Stable	Latest Release	Licenses
com.android.support:design (/maven/com.android.support:design)	runtime	23.1.1		
com.jaredrummler:android-de... (/maven/com.jaredrummler:android-device-names)	runtime	1.1.0		

core/bezirk/build.gradle (https://github.com/Bezirk-Bosch/Middleware/blob/master/core/bezirk/build.gradle)

Kind	Requirements	Latest Stable	Latest Release	Licenses
com.intellij:annotations (/maven/com.intellij:annotations)	runtime	12.0	9.0.4	12.0	Apache-2.0 (/licenses/Ap 2.0)

core/bezirk-middleware-api/build.gradle (https://github.com/Bezirk-Bosch/Middleware/blob/master/core/bezirk-middleware-api/build.gradle)

Kind	Requirements	Latest Stable	Latest Release	Licenses
com.google.code.gson:gson (/maven/com.google.code.gson:gson)	runtime	2.7	2.8.6	2.8.6	Apache-2.0 (/licenses/Apache-2.0) ⚠
org.slf4j:slf4j-api (/maven/org.slf4j:slf4j-api)	runtime	1.7.12	1.7.28	2.0.0-alpha0	MIT (/licenses/MIT) ⚠

core/commons/build.gradle (https://github.com/Bezirk-Bosch/Middleware/blob/master/core/commons/build.gradle)

Kind	Requirements	Latest Stable	Latest Release	Licenses
com.google.zxing:core (/maven/com.google.zxing:core)	runtime	3.2.0	3.3.3	3.3.3	Apache- (/licenses 2.0)
org.slf4j:slf4j-api (/maven/org.slf4j:slf4j-api)	runtime	1.7.6	1.7.28	2.0.0-alpha0	MIT (/licenses

Explore (/explore)

- Platforms (/platforms)
- Languages (/languages)
- Licenses (/licenses)
- Help Wanted (/explore/help-wanted)
- First Pull Request (/explore/first-pull-request)
- Unlicensed Packages (/explore/unlicensed-libraries)
- Trending Repositories (/github/trending)

Experiments (/experiments)

- Bus Factor (/experiments/bus-factor)
- Digital Infrastructure

About (/about)
Team (/team)
Terms and Conditions (/terms)
Privacy Policy (/privacy)
API (/api)

(/experiments/digital-infrastructure)
- Unseen Infrastructure (/experiments/unseen-infrastructure)