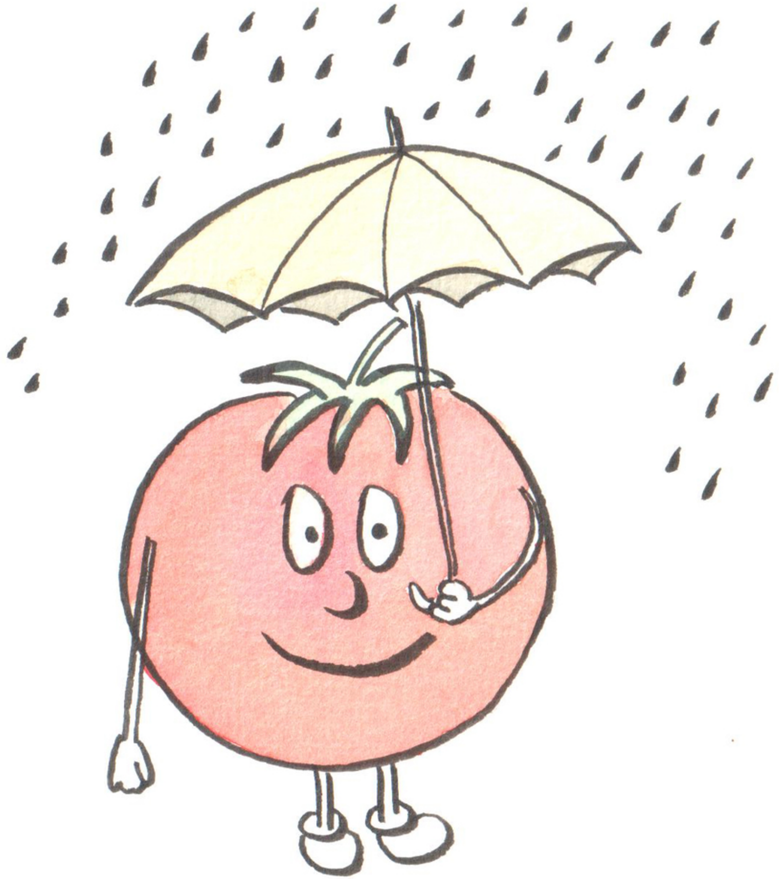
# Gartenbewässerung selbst bauen

## Eine Zeitreise mit Lötkolben, C, Java und Kotlin

Michal Harakal, 2024

Michal Harakal



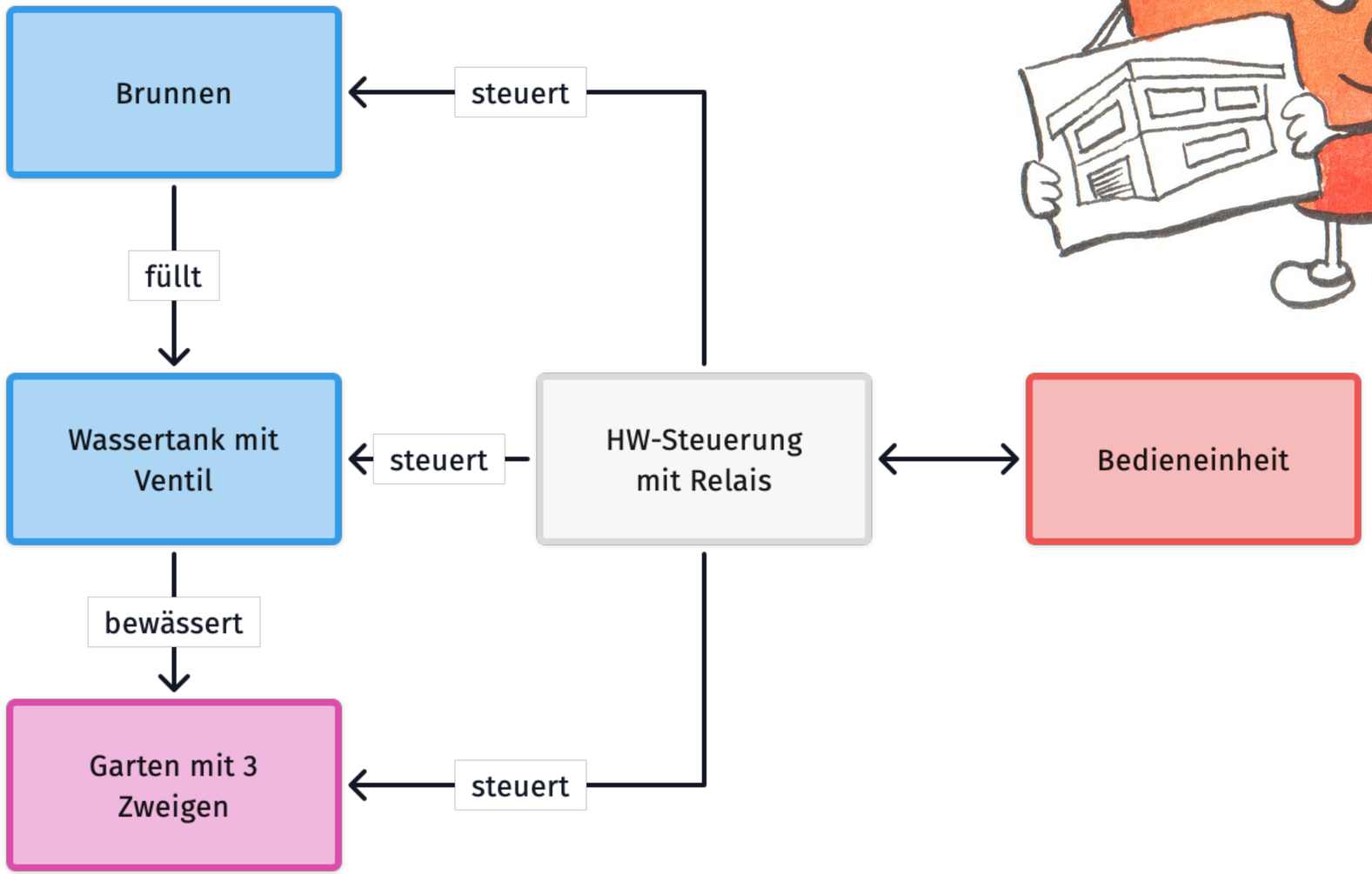
Tommy

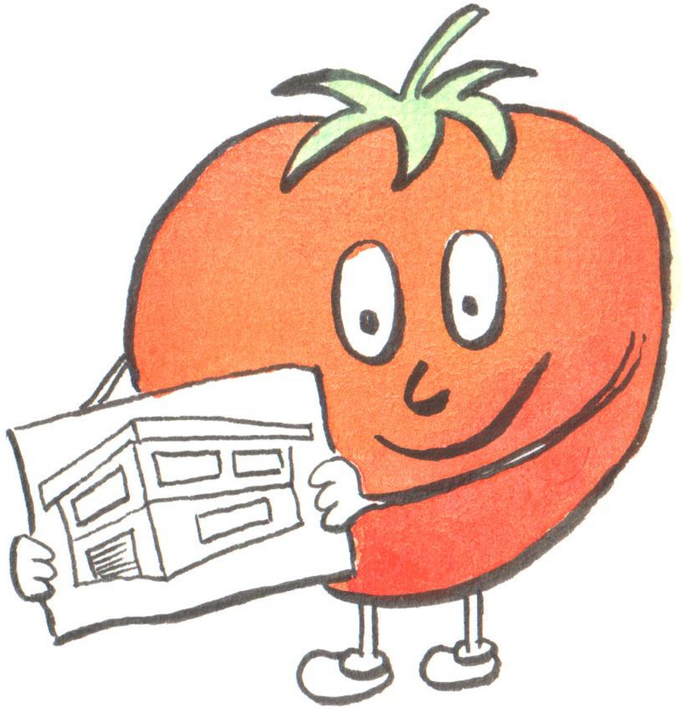
Sketch by Adriana Harakalova, 2024

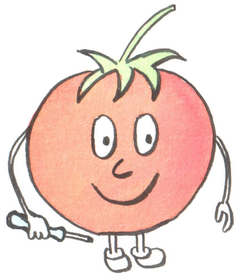
# Demeter

# Demeter

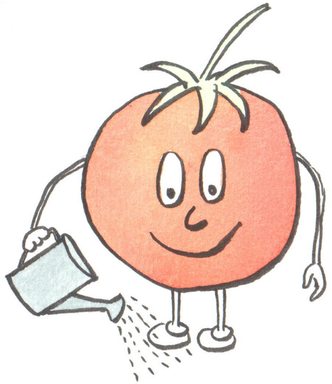
## Systemdiagram



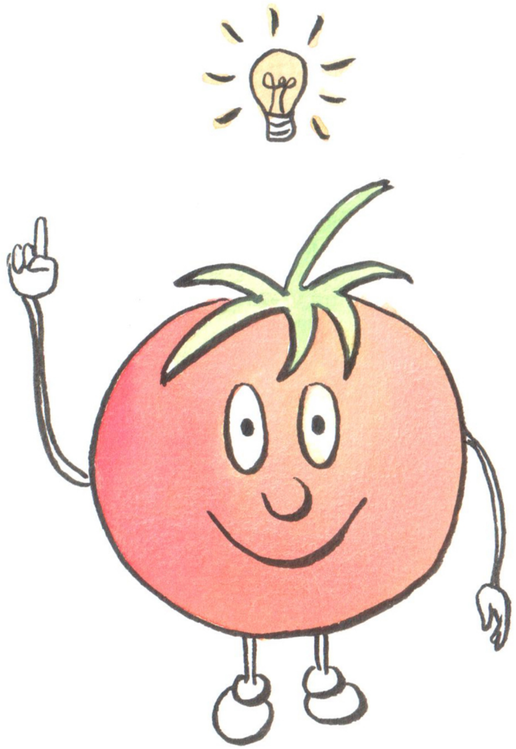




Sketch by Adriana Harakalova, 2024




GITH
VALVE
1"

Sketch by Adriana Harakalova, 2024

# Rückblick


Arduino
2005
Android OS
2007
Demeter
Cheesboard
2006


RaspberryPI
2012
Javaland
2014
Kotlin
2016
Attention is all you need
2017

androidThings

Demeter & Cloud

2018

Javaland

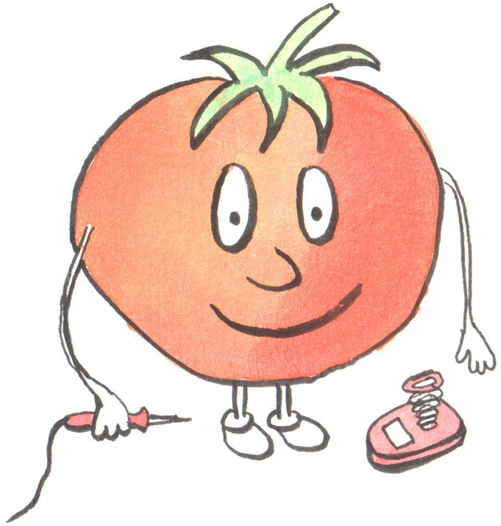
2024

Demeter

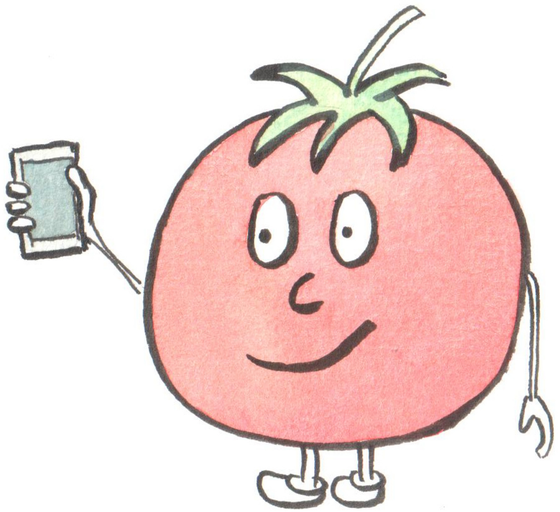
JVM App

2022

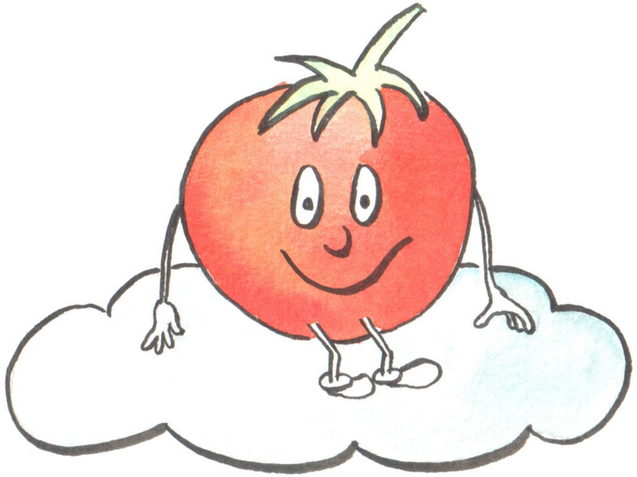
# Epochen



8-Bit RISC-Ära



Android-Ära

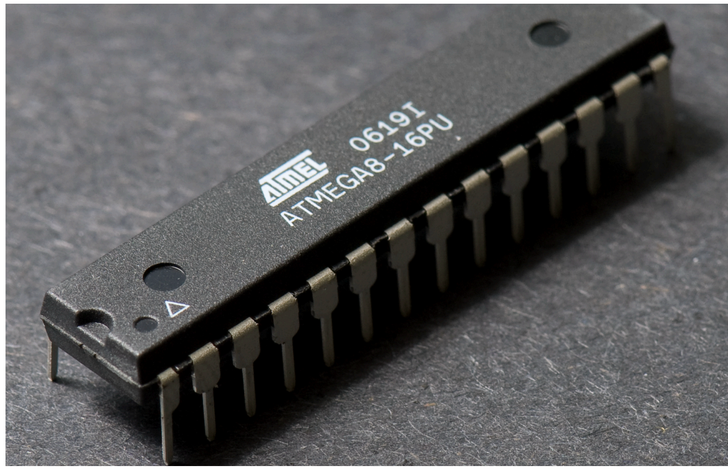


Cloud-Ära

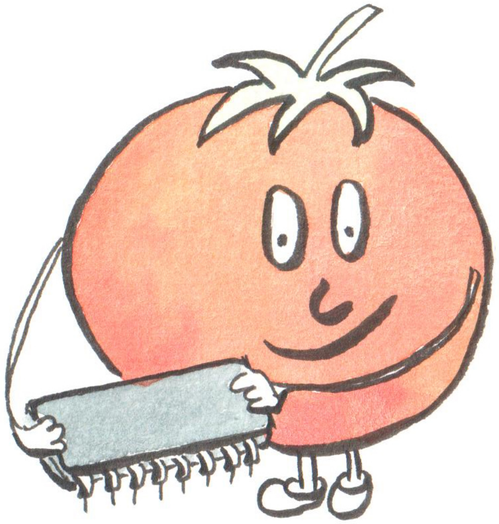
# 8-Bit RISC Ära

# ATMEGA-8

## Atmel AVR



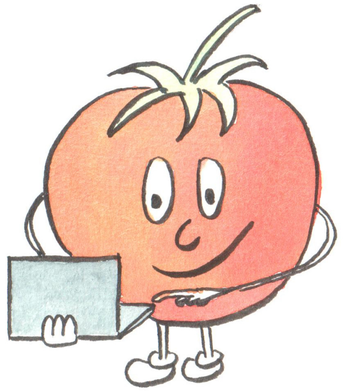

- RISC Architektur
- Bastler freundlich
- ISP (AVR ISP, In-System Programmer)
- Open Source Software (gcc & toolchain)



# ATMEGA-8 vs Apollo 11

| | Einführungsjahr | Prozessor | Taktfrequenz | Speicher | Programmiersprache |
|---|---|---|---|---|---|
| ATmega8 | Anfang der 2000er | AVR 8-Bit RISC Einzelchip-Mikrocontroller | Bis zu 16 MHz | • 8KB Flash-Speicher für Programme<br>• 1KB EEPROM<br>• 1KB internes SRAM | • C, C++<br>• Assembler |
| Apollo-Rechner (AGC) | Mitte der 1960er | Spezialgebaut, mit diskreten Logikgattern, 16-Bit Wortlänge | Ca. 1 MHz | • 36KB (ROM)<br>• 2KB RAM | Programmiert in AGC-Assembler |

# Arduino



Blink | Arduino IDE 2.3.2

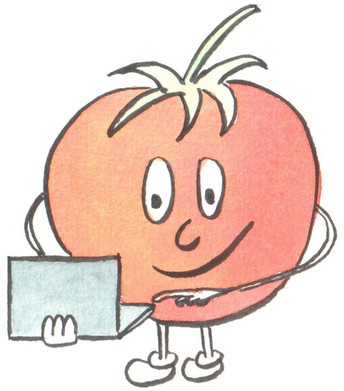
Arduino Nano 33 BLE

Blink.ino

```
/*
  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Ln 1, Col 1  Arduino Nano 33 BLE on /dev/cu.usbmodem1101 [not connected]

# Mainloop



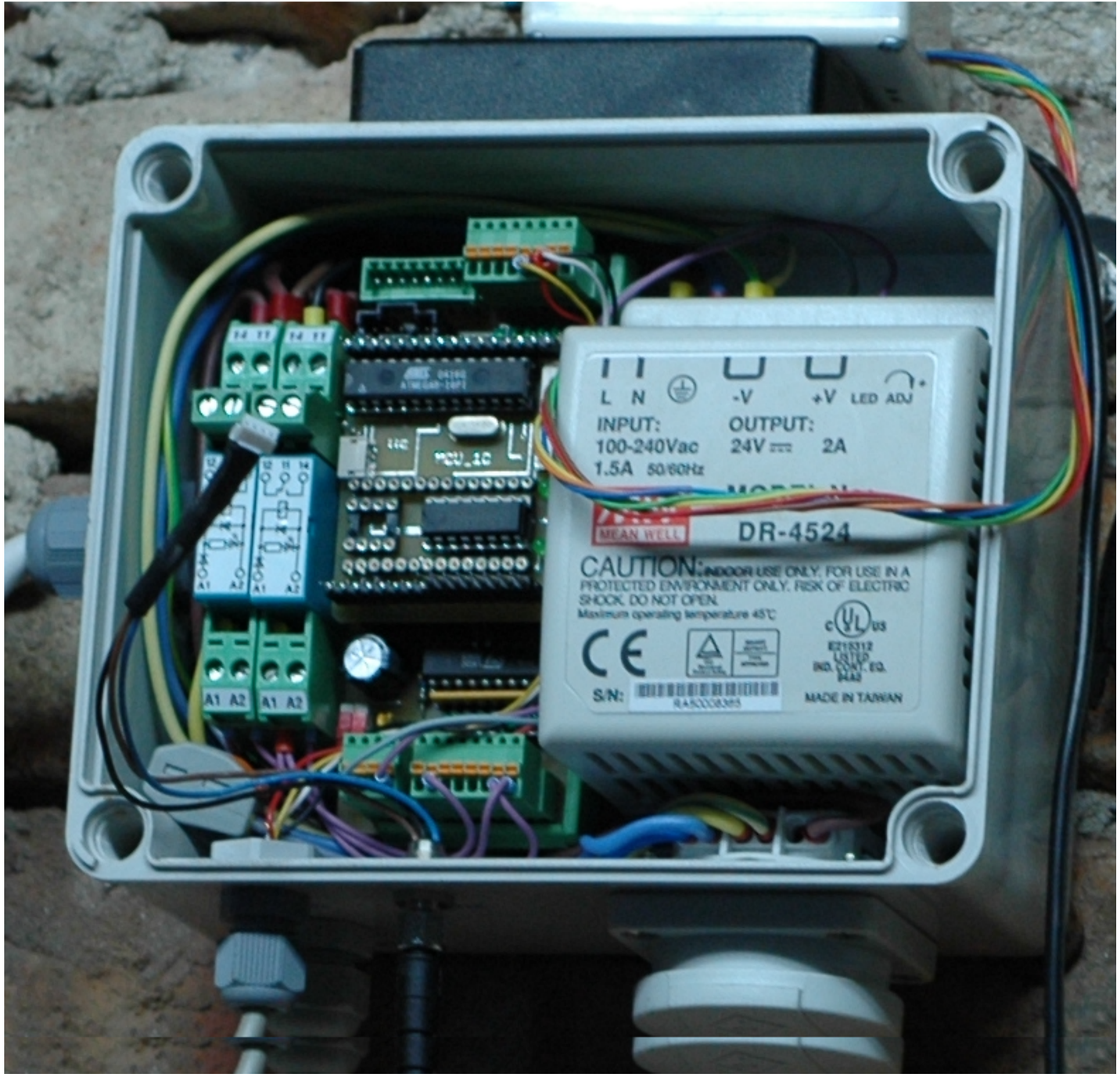
scheduling_loop | Arduino IDE 2.3.2

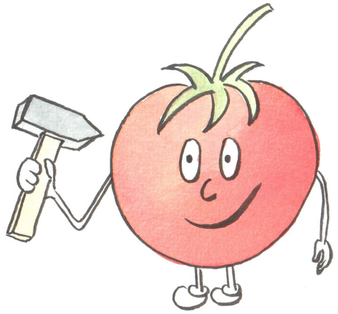
Arduino Nano 33 BLE

scheduling_loop.ino

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  unsigned long currentMillis = millis();

  // Check if a second has passed
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;  // Save the last time you blinked the LED

    // If the LED is off, turn it on, and vice-versa:
    int ledState = digitalRead(LED_BUILTIN);
    digitalWrite(LED_BUILTIN, !ledState);

  }
  // Check in every loop
  checkSwimmSensor();
}

void checkSwimmSensor() {
  // code checks for swimm sensor and closes valve after overflow
}
```

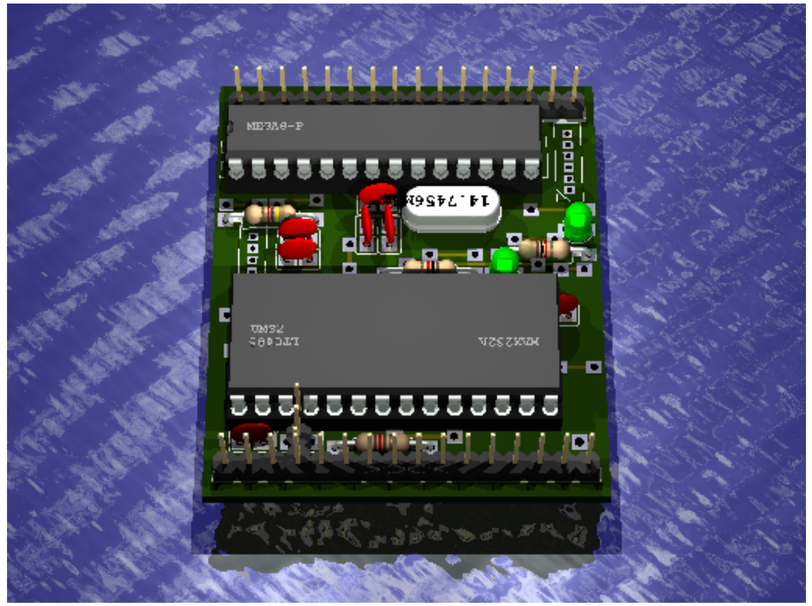
Ln 22, Col 66  Arduino Nano 33 BLE on /dev/cu.usbmodem1101 [not connected]

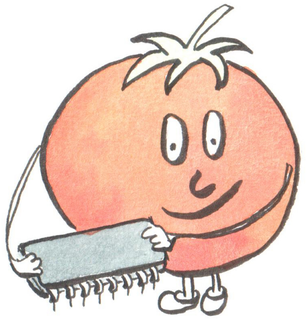


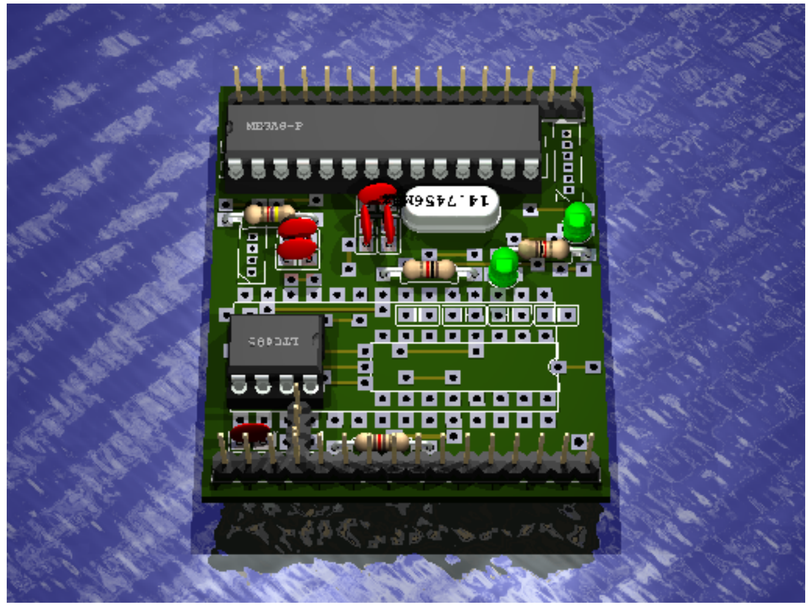



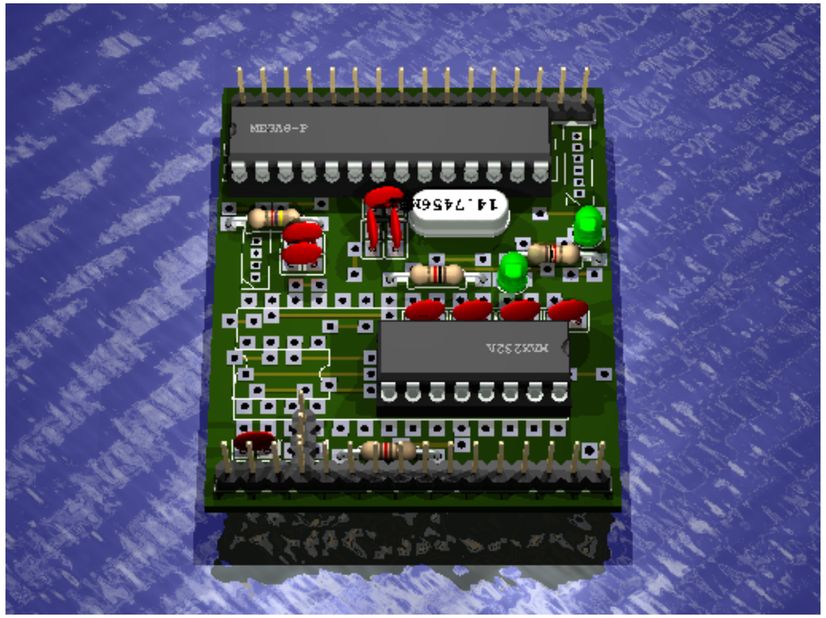
Sketch by Adriana Harakalova, 2024

# Cheesboard

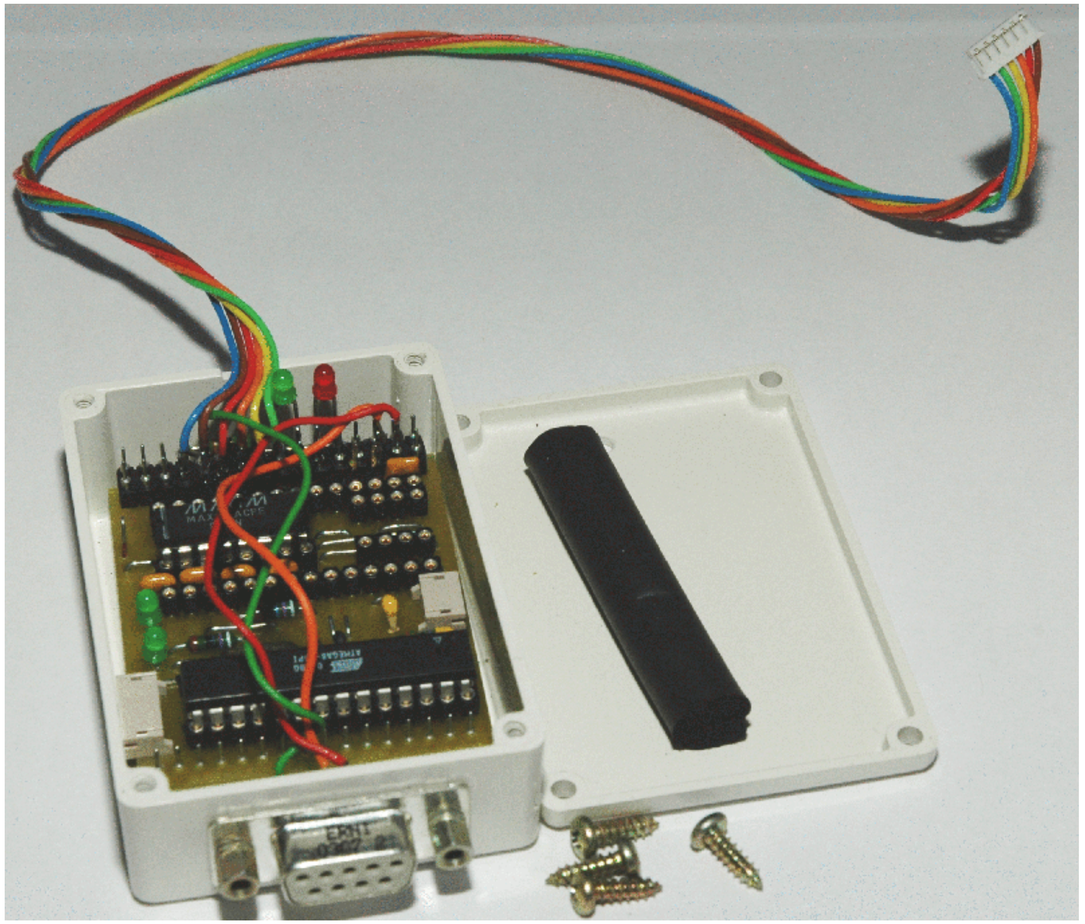


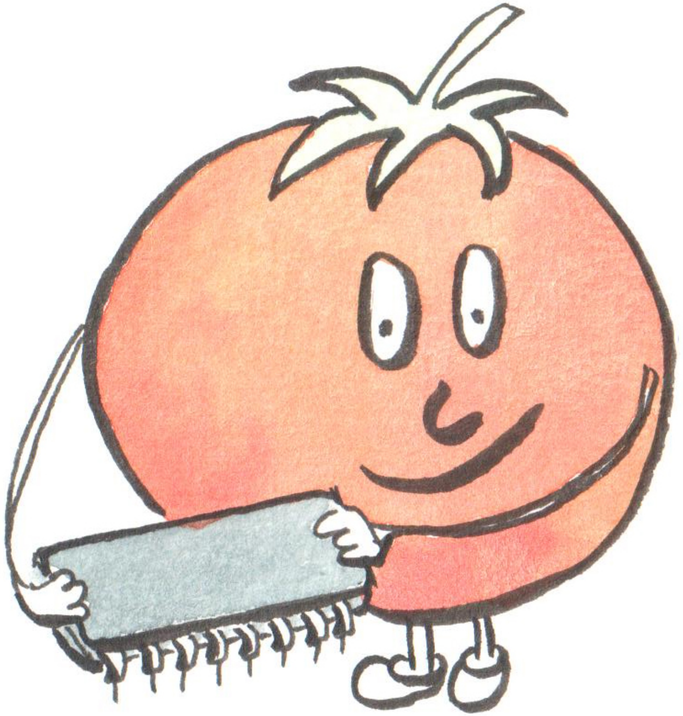






Sketch by Adriana Harakalova, 2024





Sketch by Adriana Harakalova, 2024

# Cheesboard

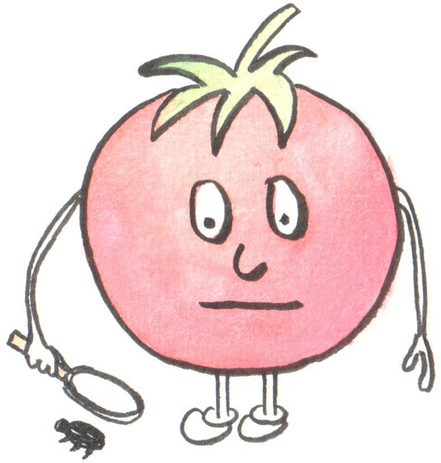
## Als es noch das Arduino nicht gab

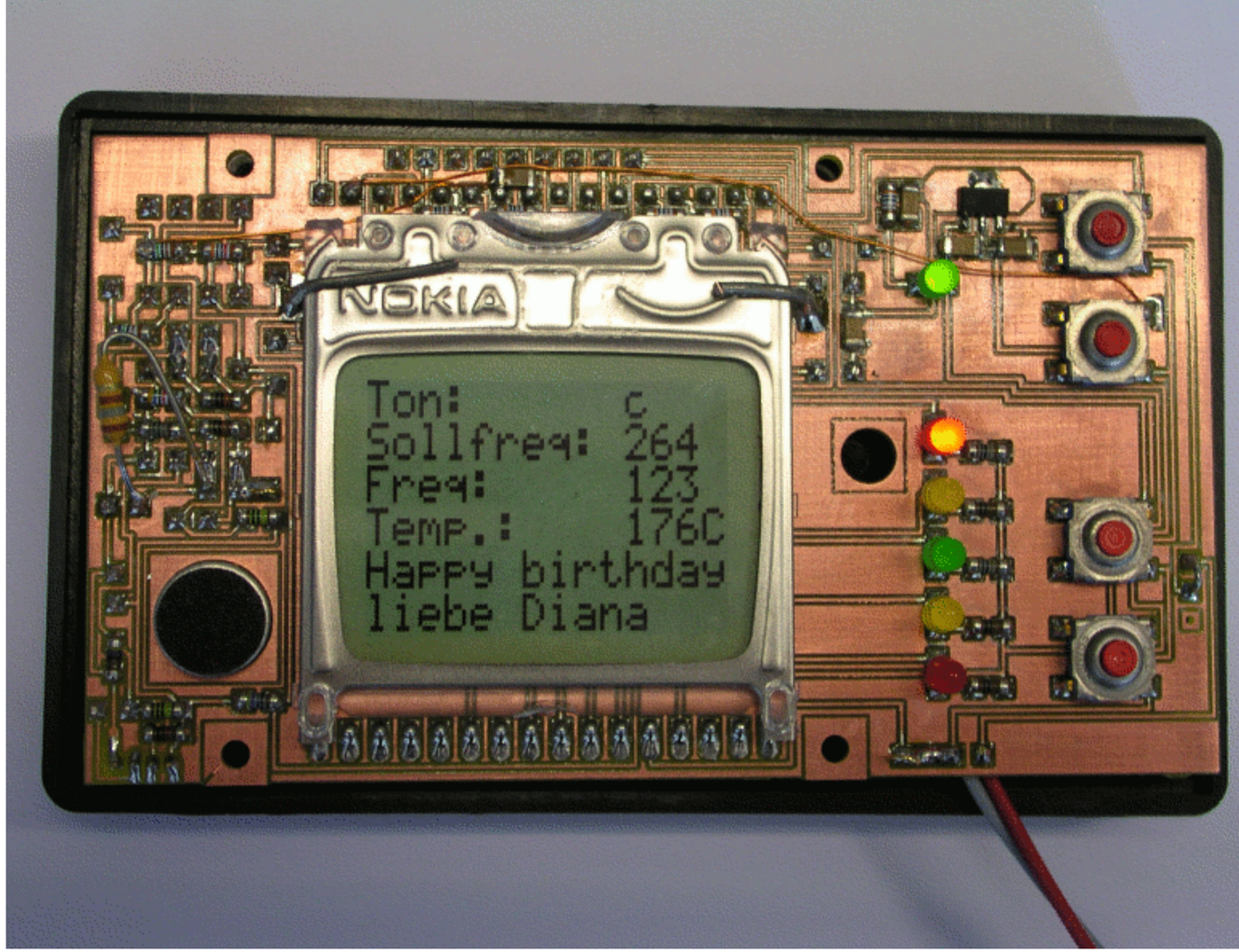
- Modulares steckbares Platinen Konzept
  - Steckbare Pinsleisten
  - konfektionierte Kabelverbinder
- Module
  - Display (Nokia)
  - MCU
  - Kommunikation
  - Tastatur

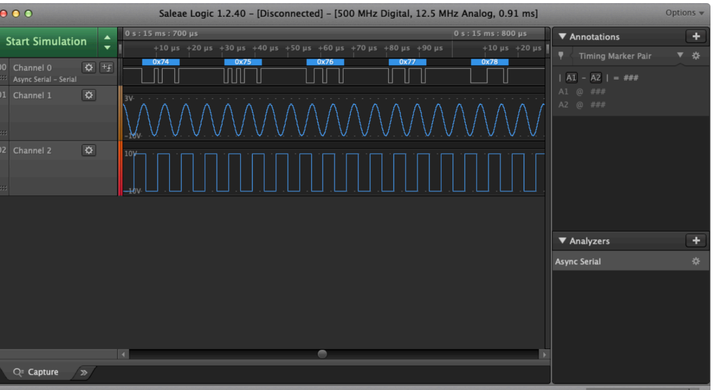
# Entwicklung

## NOKIA 6151 Display(PCD8544)

- 84X48 Pixel

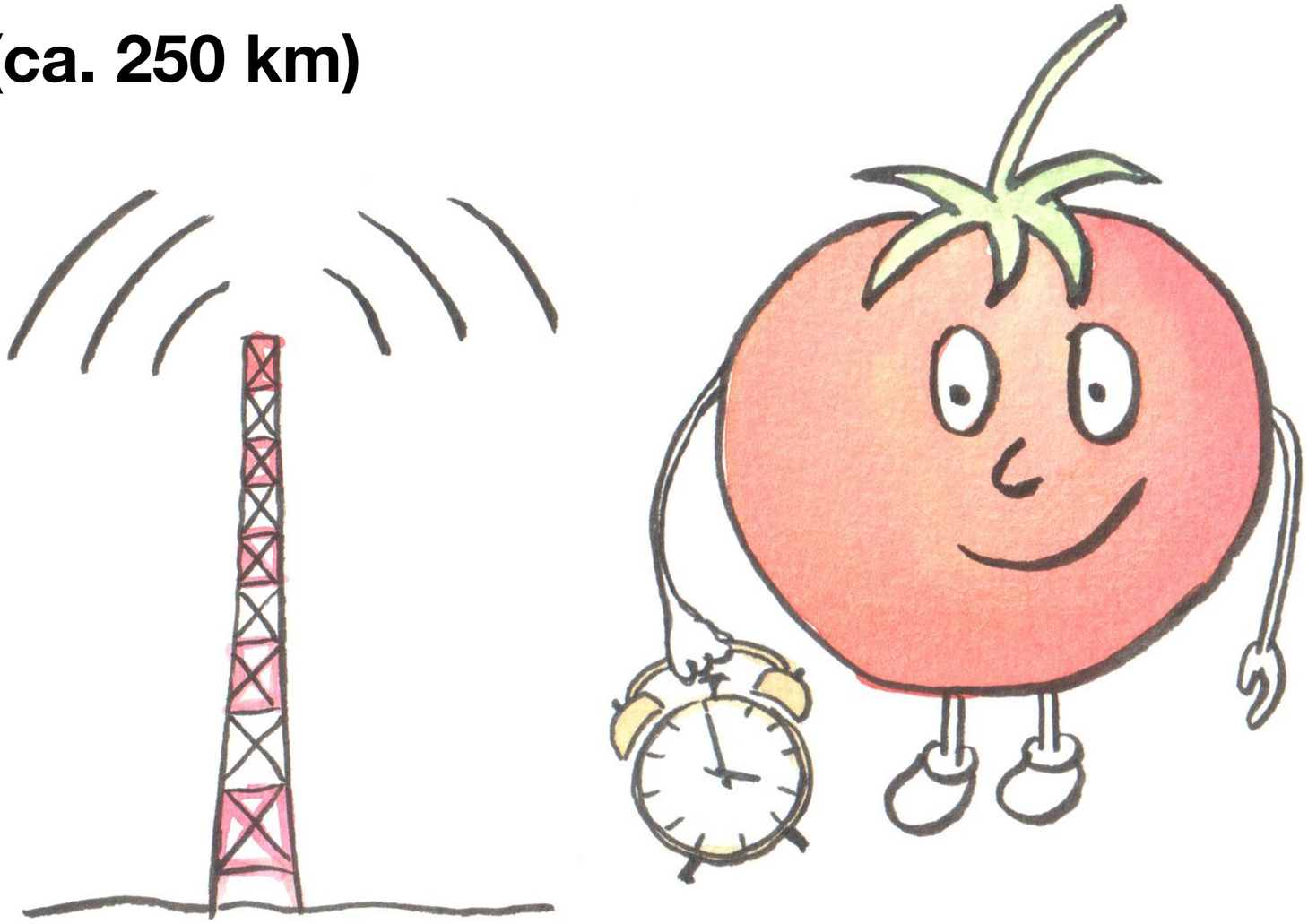




# Zeitmessung

## DCF-77 Empfänger (ca. 250 km)



# Android Ära

# Android things

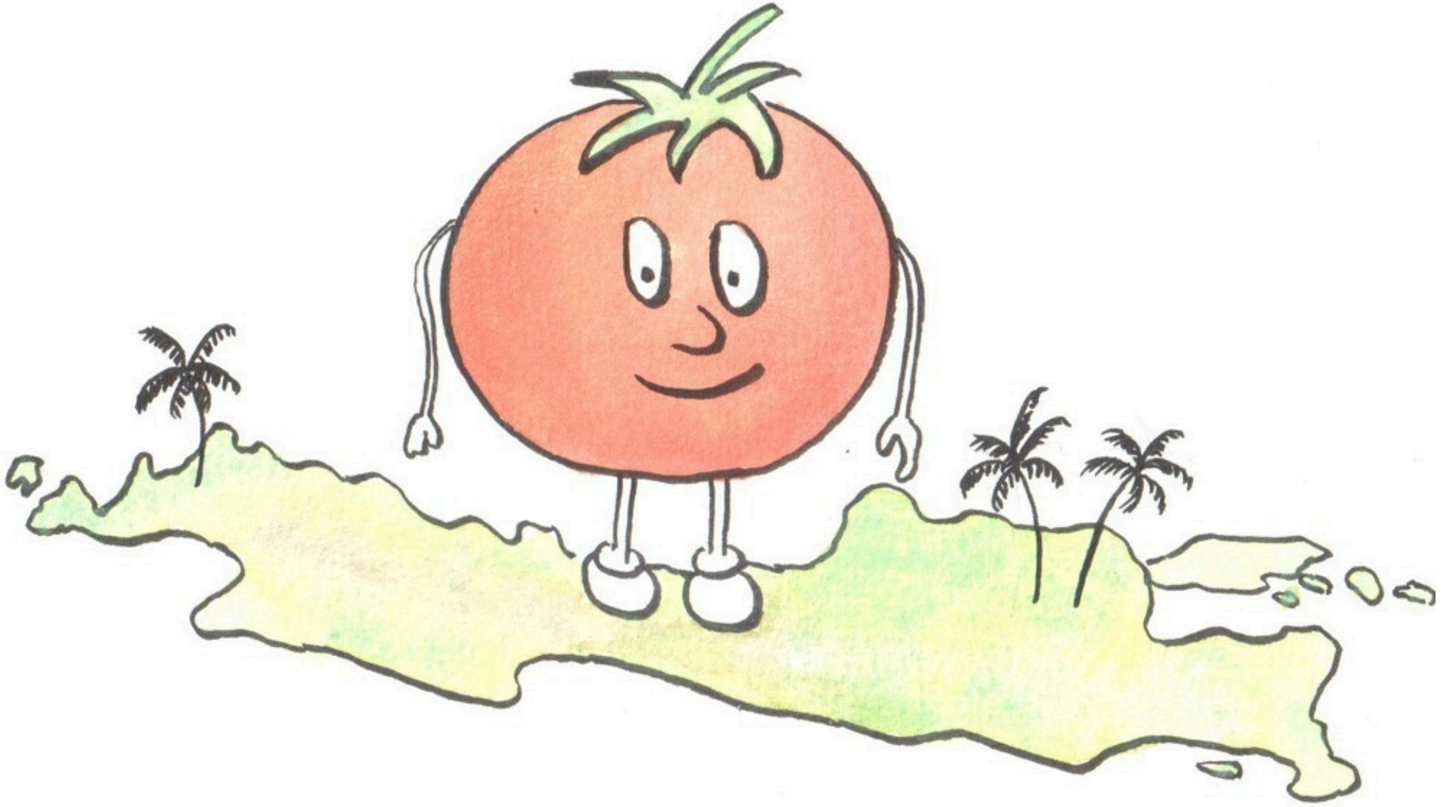
## Android-based embedded operating system

- Low power und Internet Of Things
- Over The Air Updates
- Geräte Management
- Betriebssystem Image für RaspberryPi
- Android

androidthings

# Android & Java

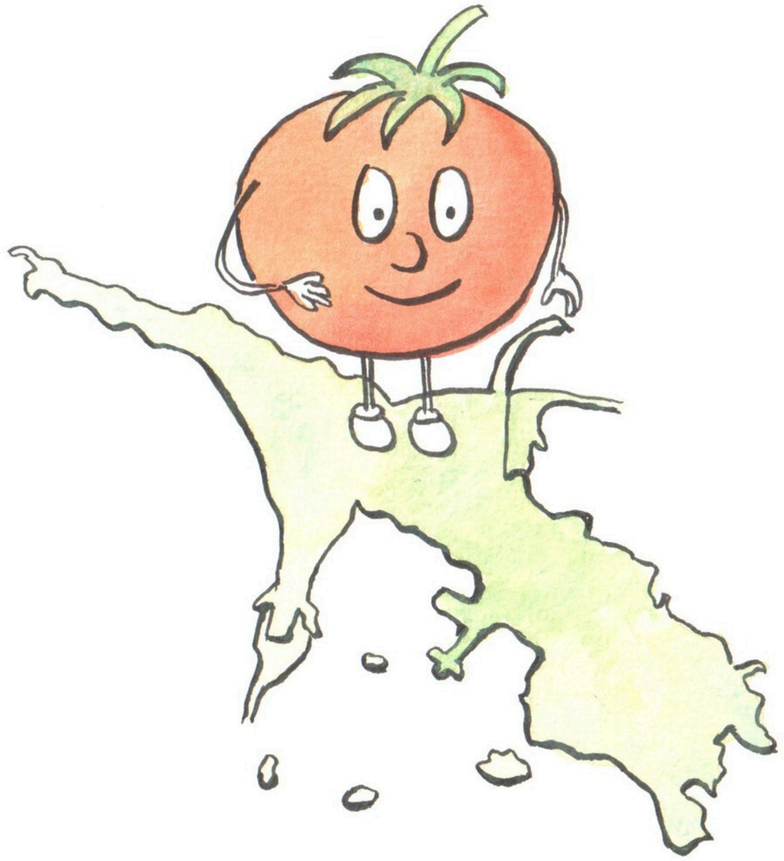
- Java 8
- Gradle builds
- Treiber für Hardware
- RaspberryPi als „Android Device"



# Android & Kotlin

## Kotlin ist auch eine Insel

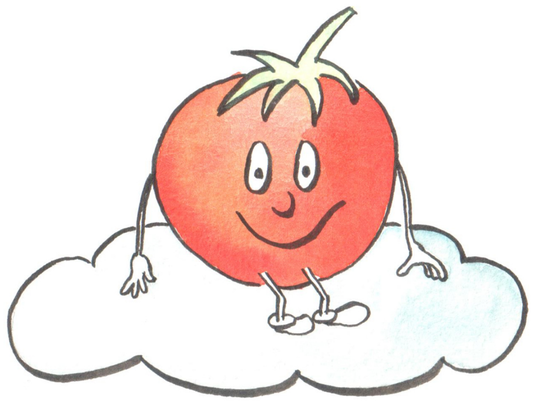
- 2016 von Google offiziell unterstützte Sprache
- Kotlin Multiplatform
- Compose Multiplatform

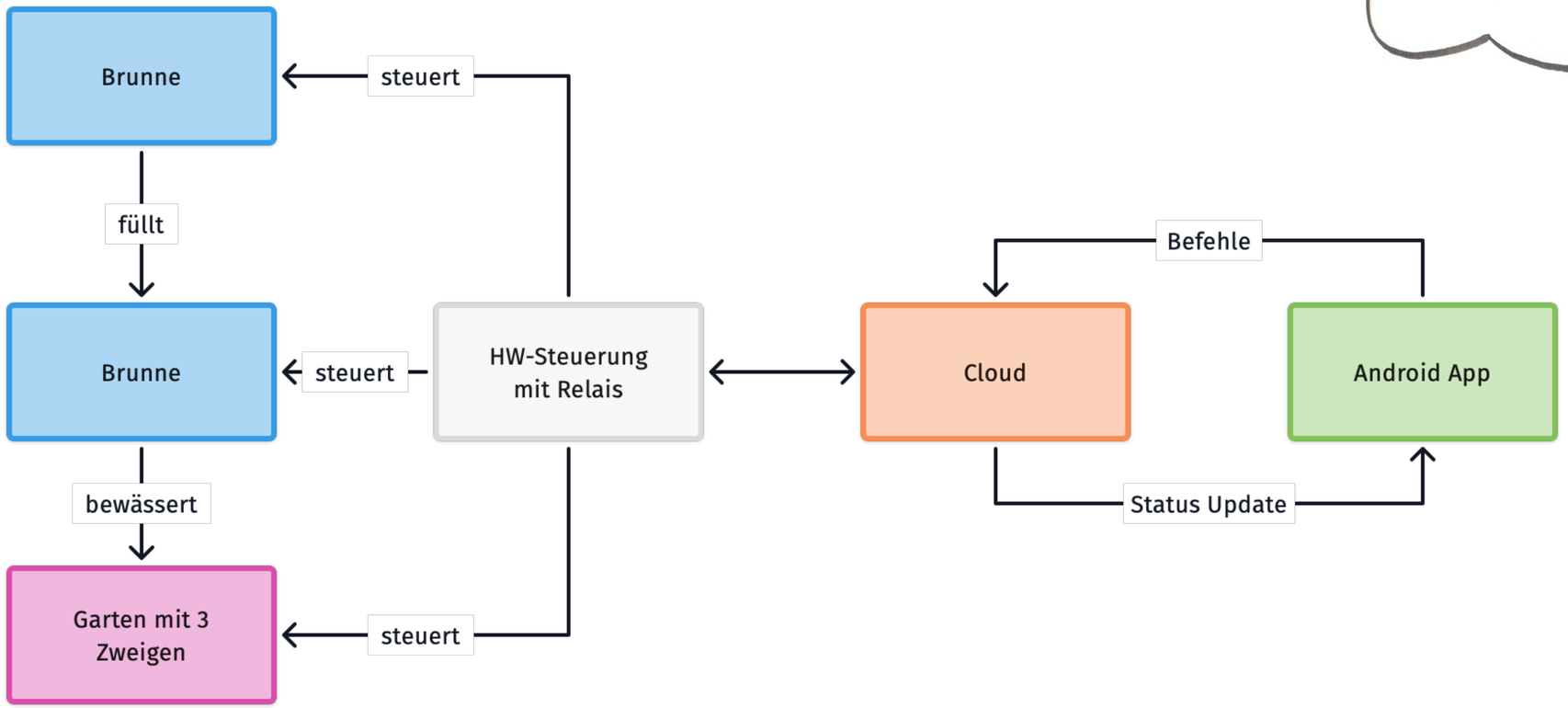


# Cloud Ära

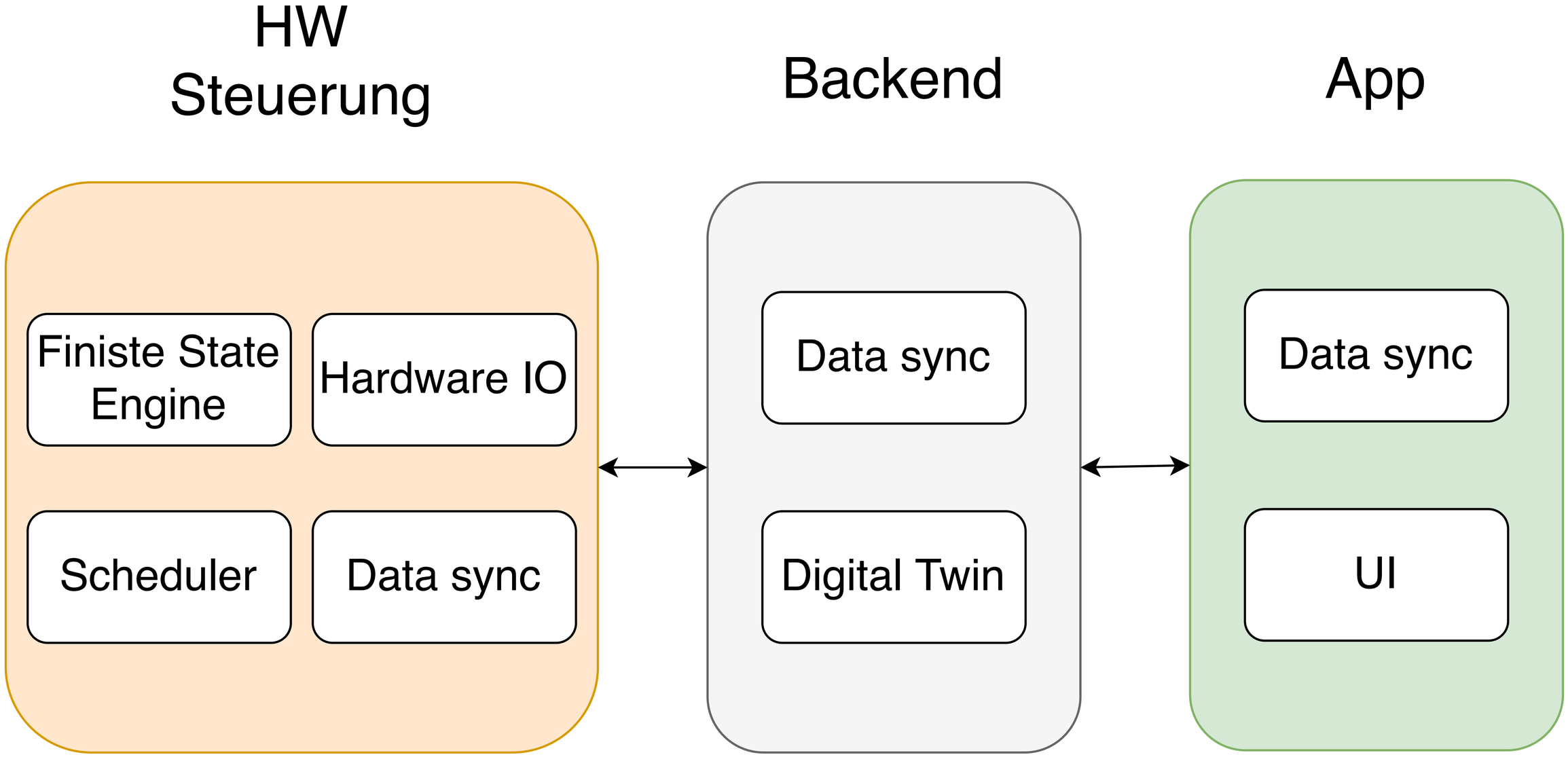
# Demeter in der Wolke

## Systemdiagram



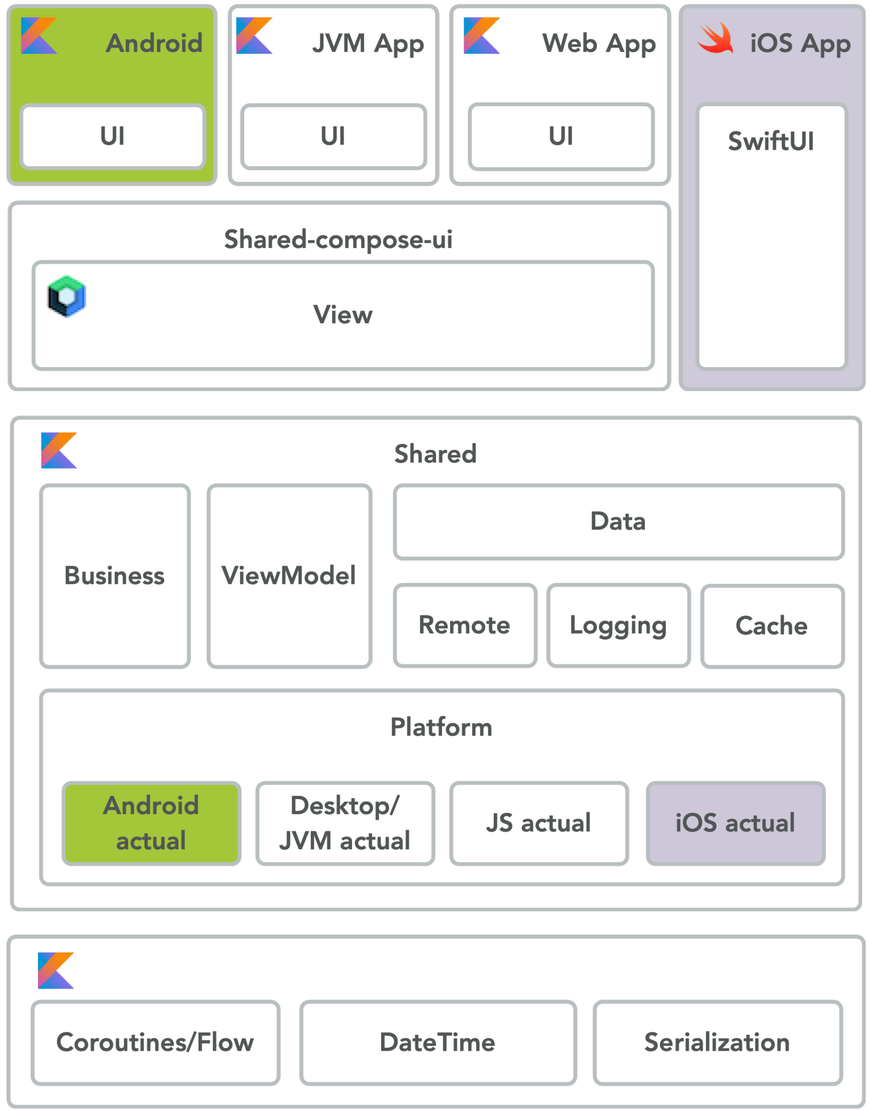


# Demeter in der Cloud



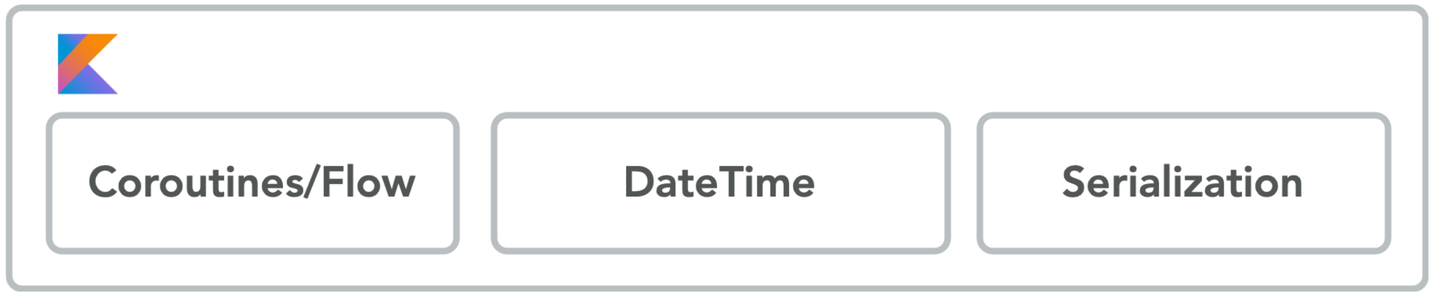
# App

# ARCHITEKTUR



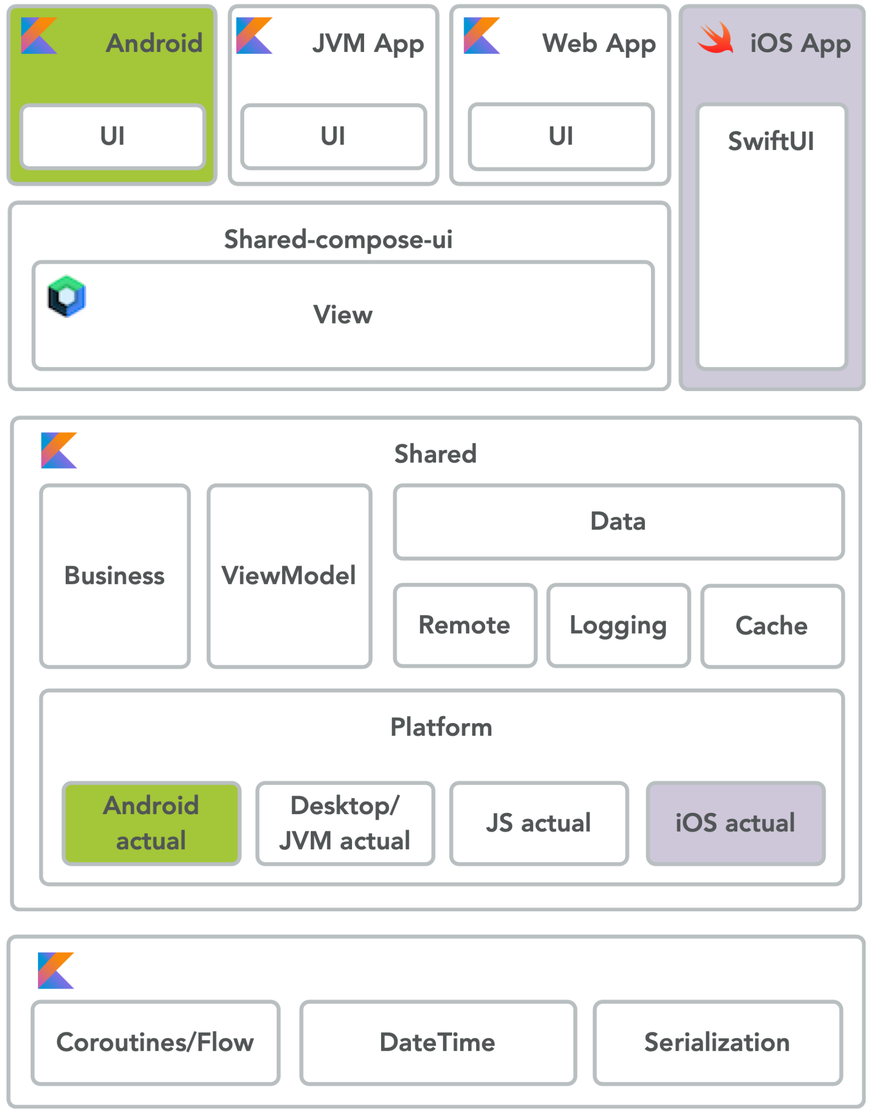
# ARCHITEKTUR

## FOUNDATION



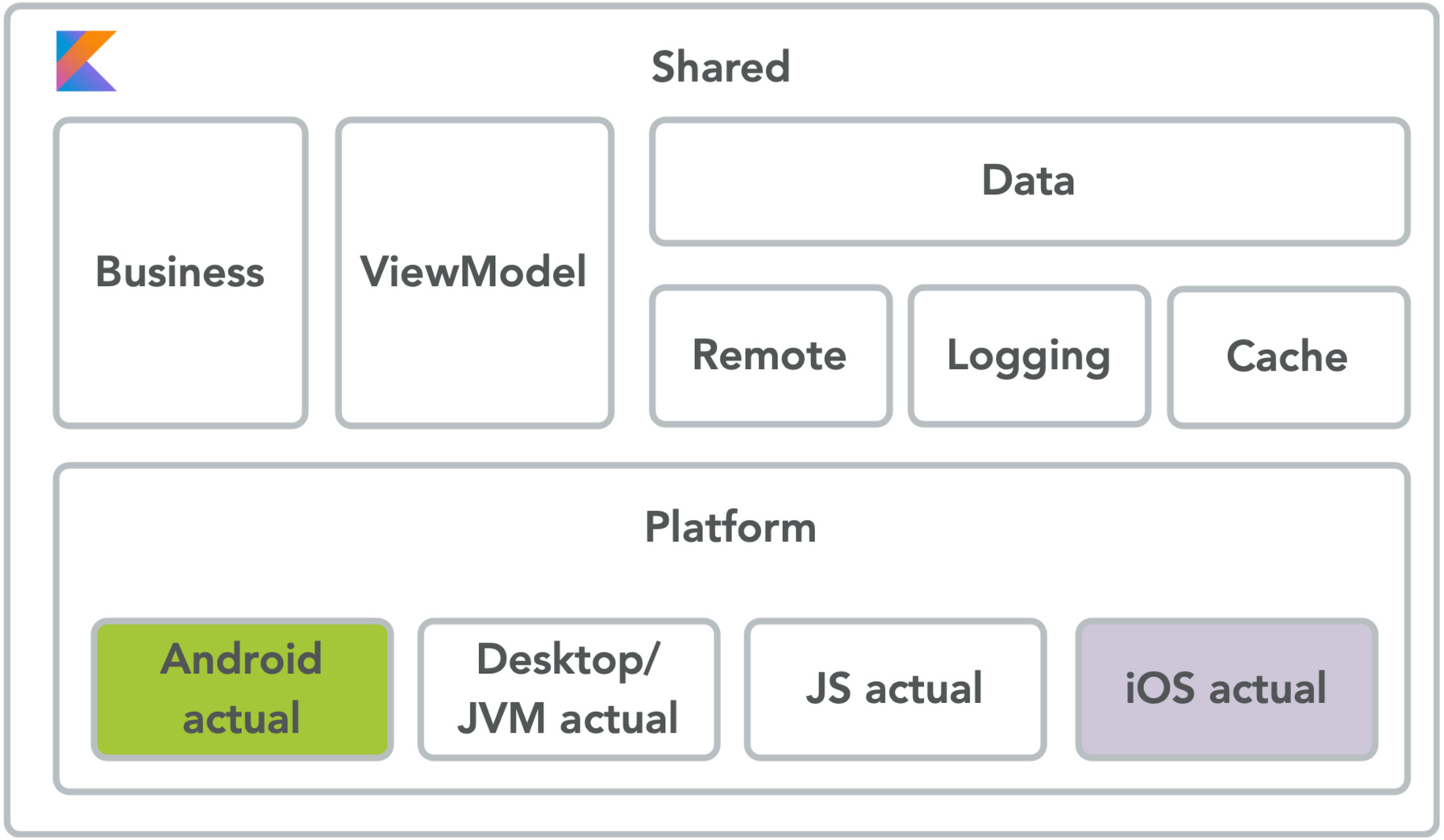

- Kotlin Standard Library
- Kotlin/Native
- Kotlinx libraries

# ARCHITEKTUR

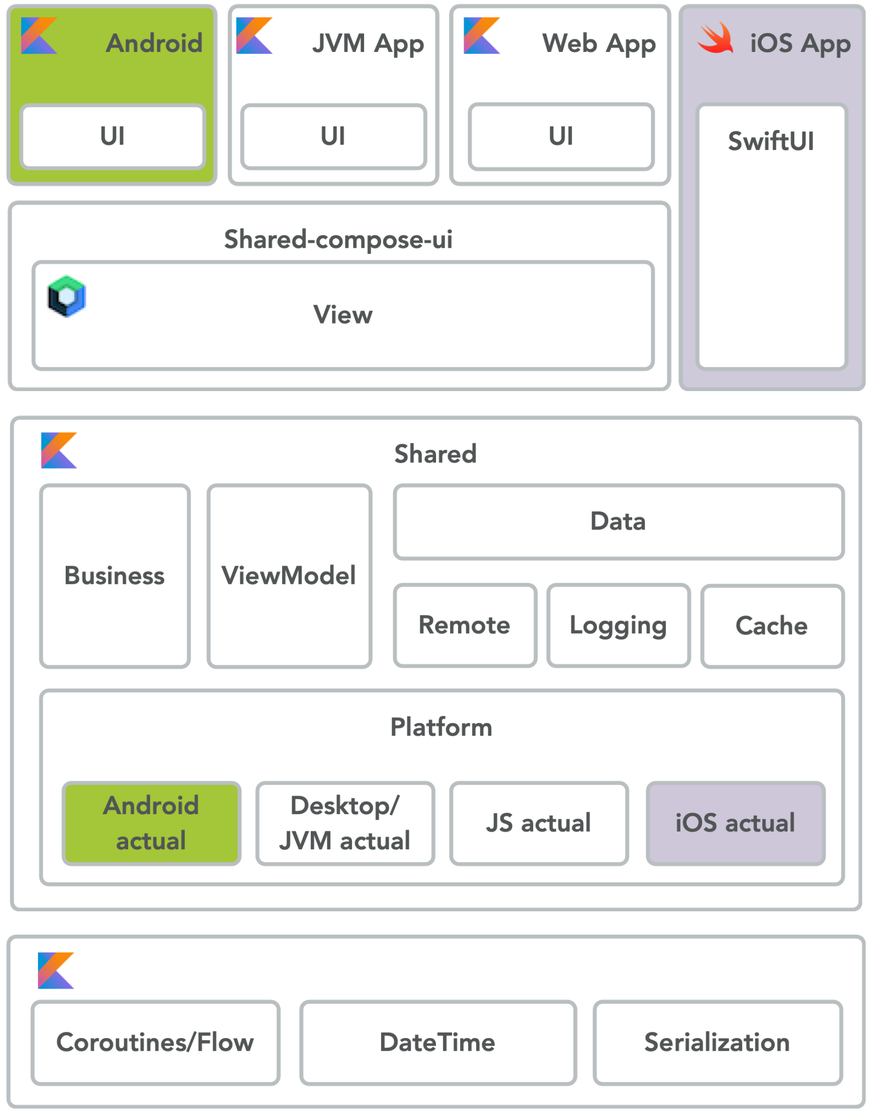


# ARCHITECTURE

## Shared Module

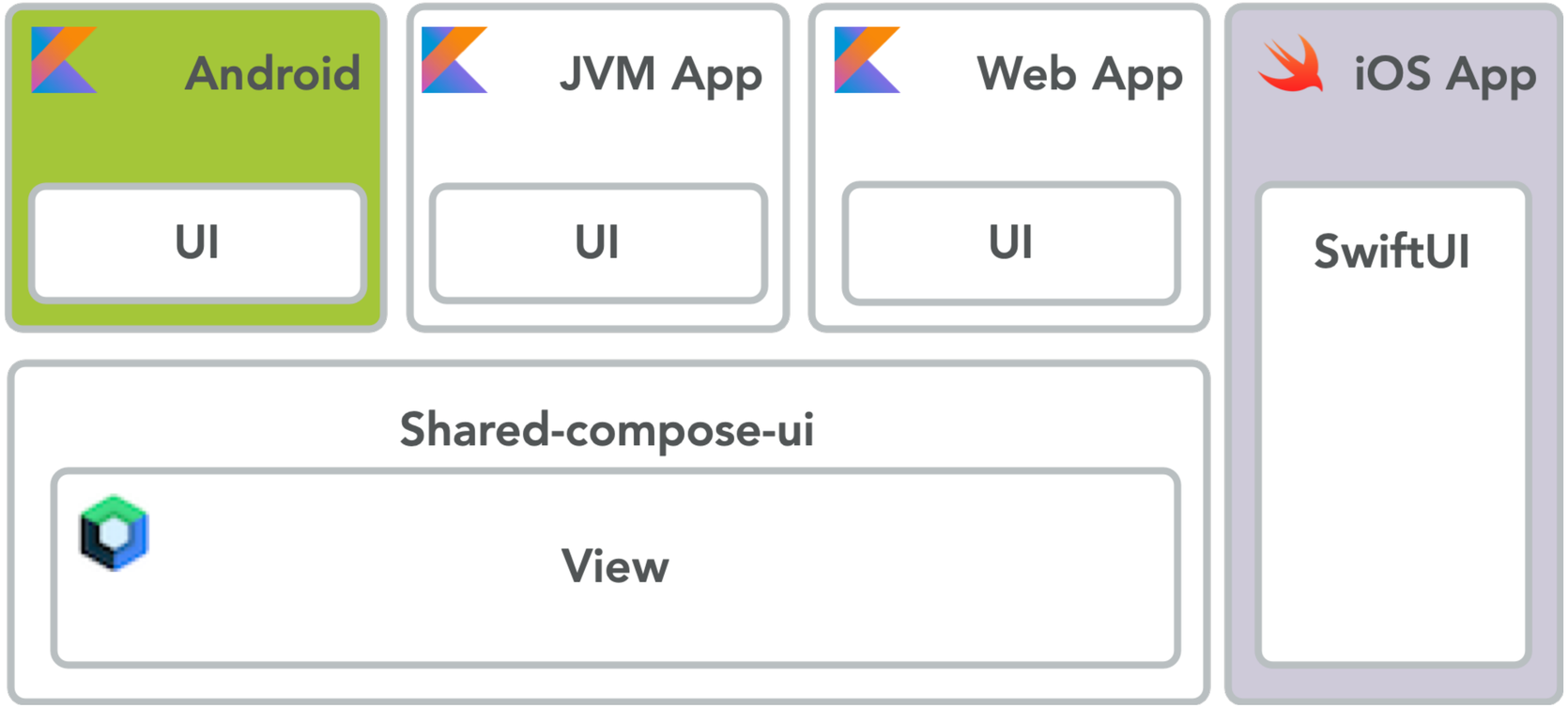


# ARCHITECTURE



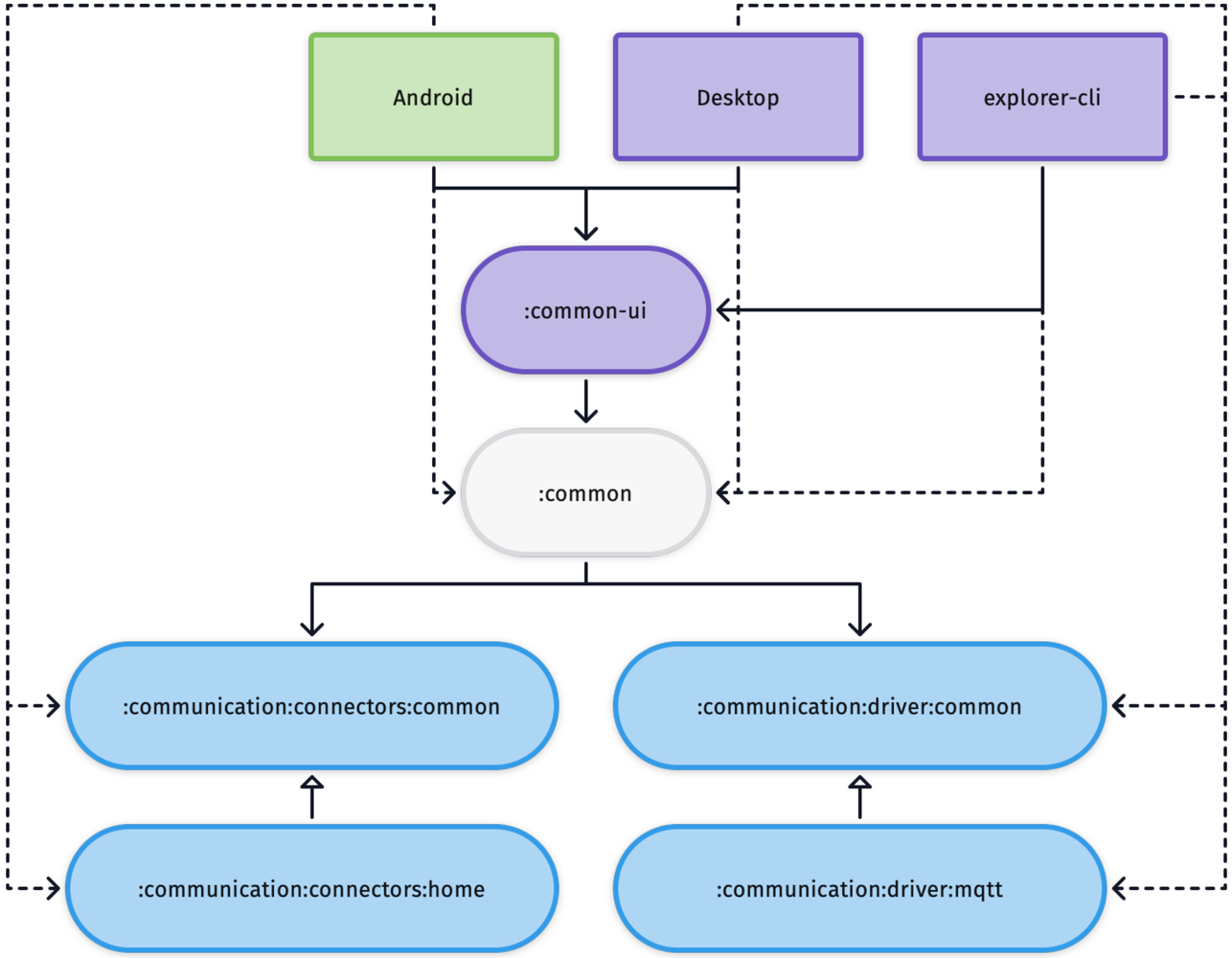
# ARCHITEKTUR

## APPS



# GRADLE MODULE

## GRADLE



# Modularisierung

## OpenAPI (aka Swagger)

develop | Files | Filter files

demeter-2018 / common / **client-api**

| Name | Size | Last commit | Message |
|---|---|---|---|
| .. | | | |
| .gitignore | 7 B | 2020-04-12 | Use common module for sharin client API logic |
| build.gradle | 1.44 KB | 2020-04-12 | Use common module for sharin client API logic |

# Modularisierung

## OpenAPI (aka Swagger)

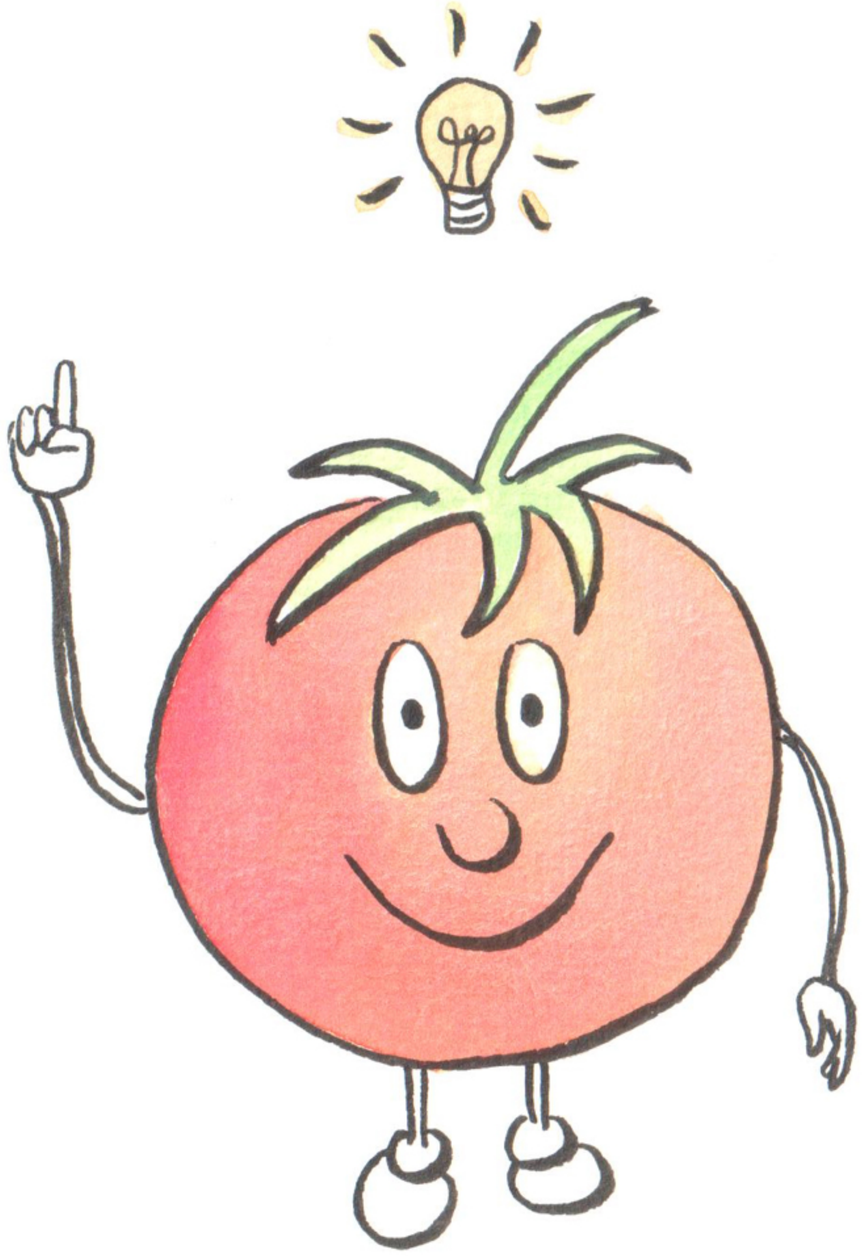
demeter-2018 / common / client-api / **build.gradle**

```
apply plugin: 'java'
apply plugin: "org.detoeuf.swagger-codegen"
apply from: "${project.rootDir}/versions.gradle"

sourceCompatibility = 1.8
targetCompatibility = 1.8

swagger {
    inputSpec = "$project.rootDir/common/specs/swagger.json"

    outputDir = "$buildDir/swagger"

    lang = 'java'

    additionalProperties = [
            'invokerPackage'   : 'com.fiwio.iot.demeter.api.service',
            'modelPackage'     : 'com.fiwio.iot.demeter.api.model',
            'apiPackage'       : 'com.fiwio.iot.demeter.api',
            'library'          : 'retrofit2',
            'serializableModel': 'true',
            'dateLibrary'      : 'joda'
    ]
}
```

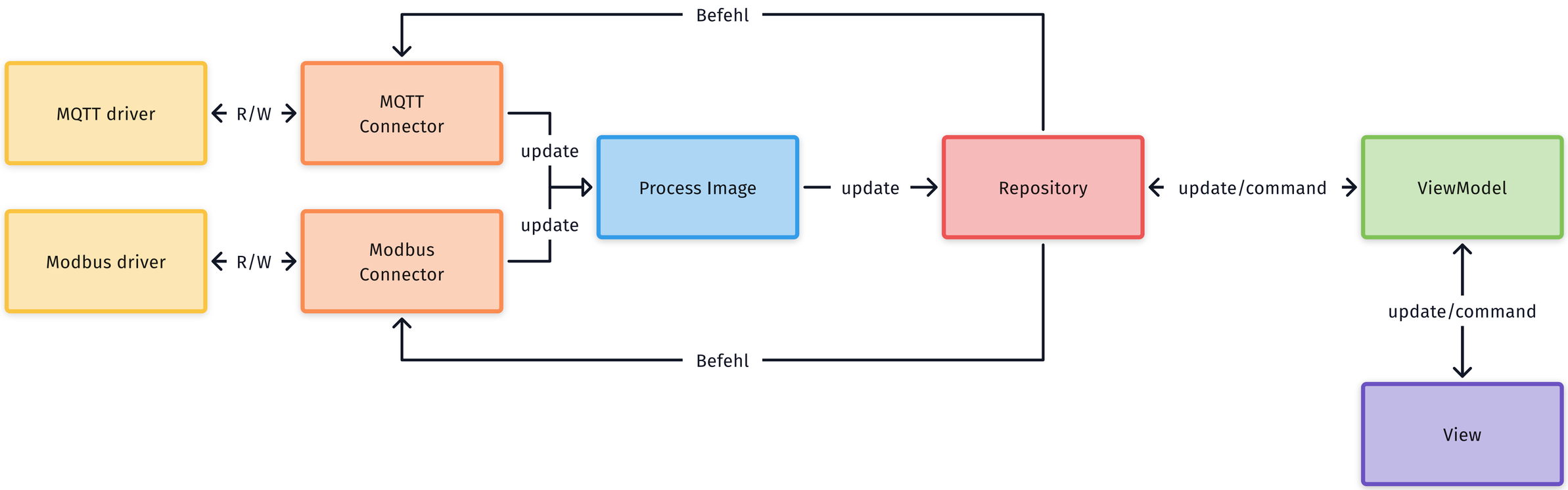
# Zukunft

# Ideen

- App verbessern
- Smarthome und Home Assisted Plattformen
- Kommerzielle Systeme



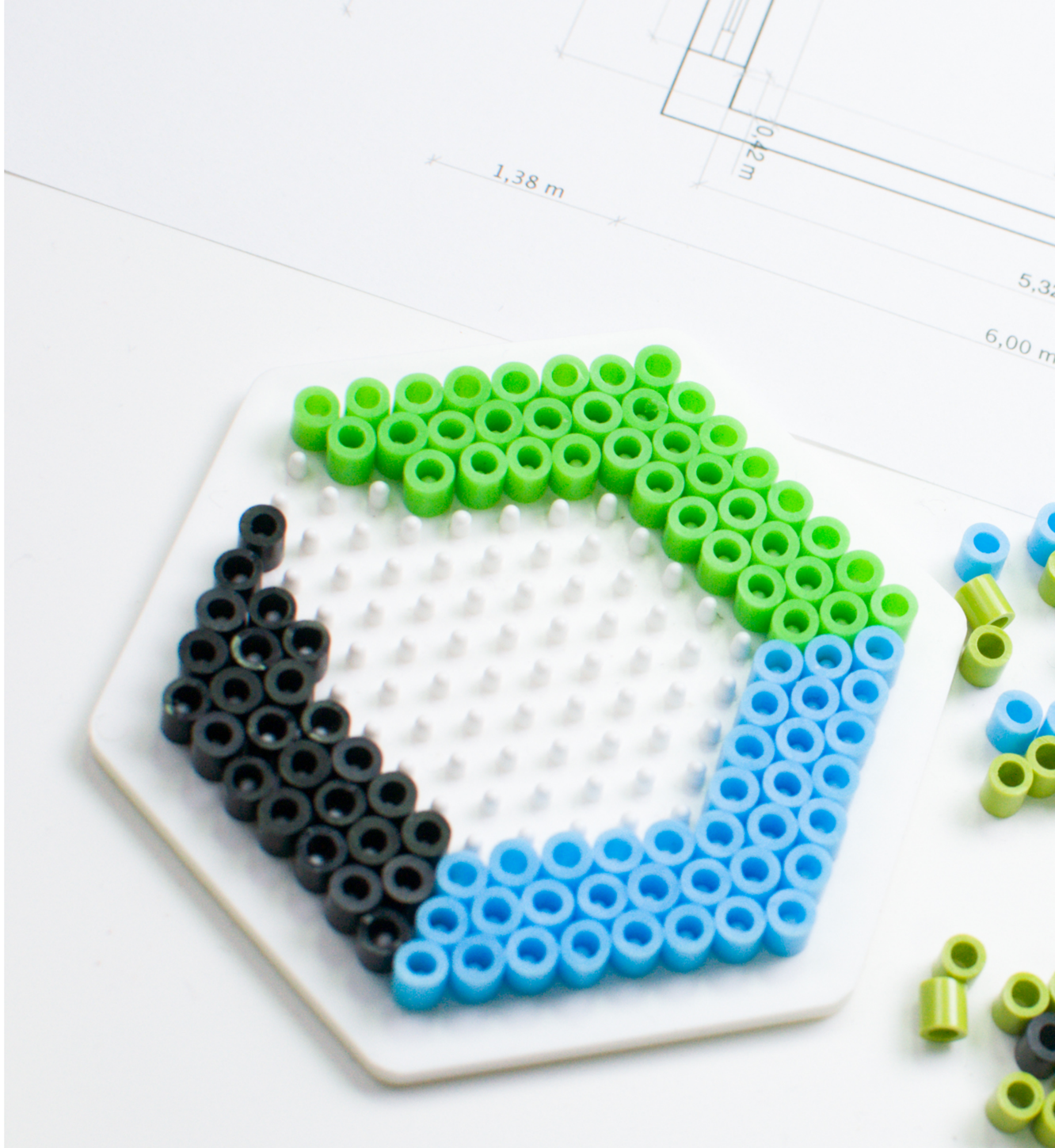
# UNIDIRECTIONAL DATA FLOW



# Jetpack Compose

## Reaktive UI

- reaktives, deklaratives UI framework
- stabil seit Juli 2021
- Mit Kotlin Compiler Plugin
- portierbar von Anfang an



# Compose Multiplatform

- JVM Desktop
- iOS
- Web

# Compose Multiplatform

## Code sharing

```
import kotlinx.coroutines.CoroutineScope

expect open class CommonViewModel() {
    val clientScope: CoroutineScope
    protected open fun onCleared()
}
```

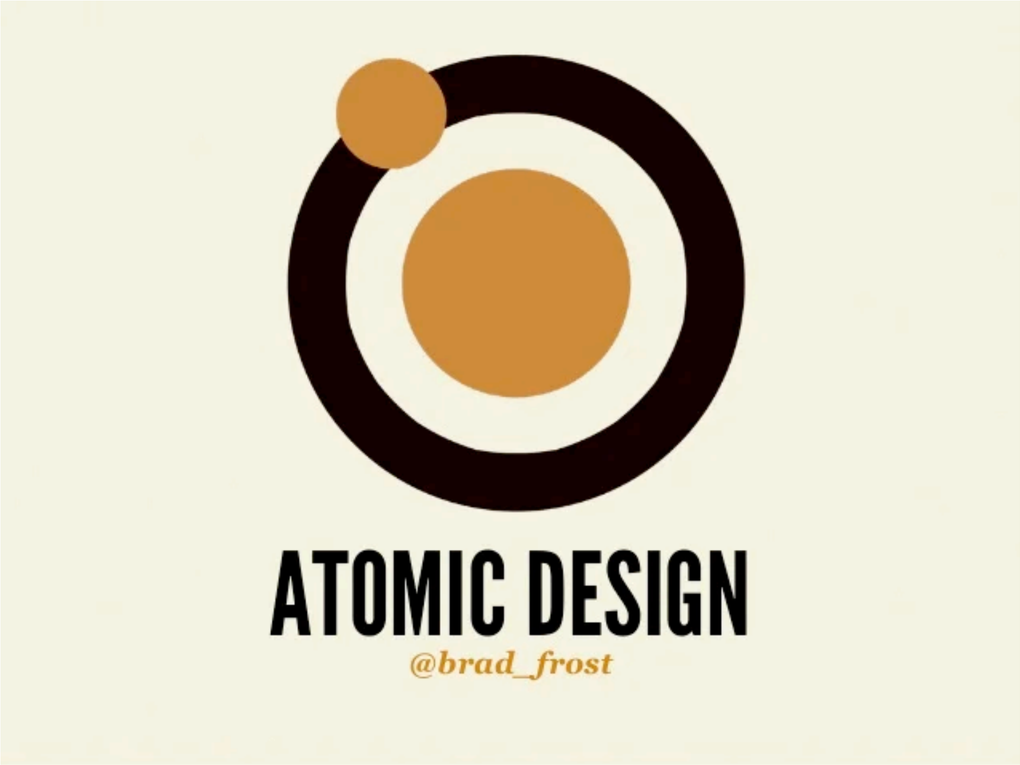
# Compose Multiplatform

## Code sharing

```
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import kotlinx.coroutines.CoroutineScope

actual open class CommonViewModel actual constructor() : ViewModel() {
    actual val clientScope: CoroutineScope = viewModelScope
    actual override fun onCleared() {
        super.onCleared()
    }
}
```

# Atomic Desing, Brad Frost

- Atom
- Molecule
- Organism
- Template
- Screen



# Atomic Design, Brad Frost

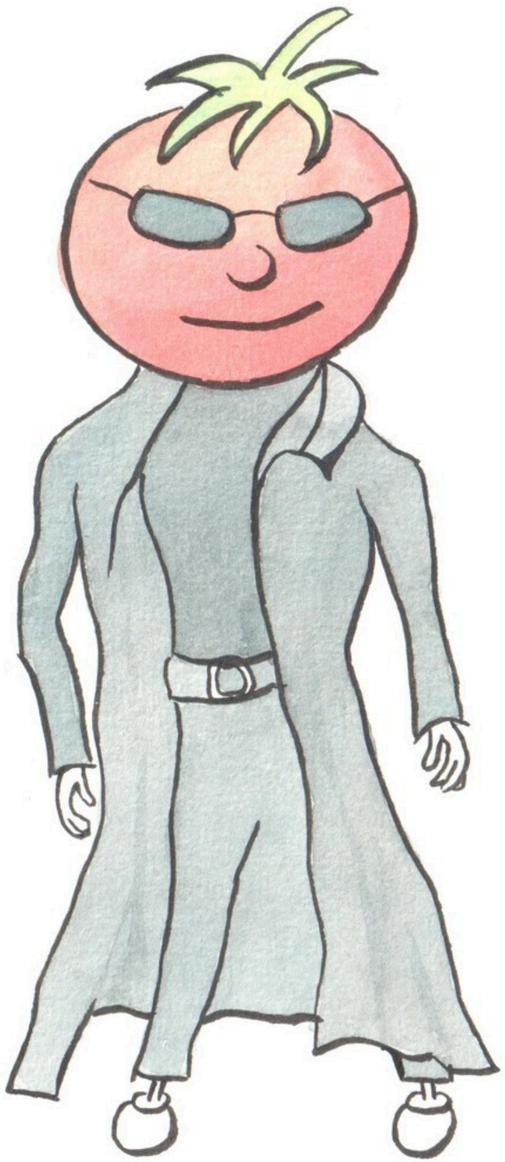
## ATOM UI ELEMENT

LED – AN/AUS


SCHALTER – AN/AUS

# DANKE

# MATRIX



# DANKE

# Links, Nachweise, Copyrights

# Links, Copyrights

- https://upload.wikimedia.org/wikipedia/commons/a/a9/ATmega8_01_Pengo.jpg
- https://de.m.wikipedia.org/wiki/Datei:Arduino_Logo.svg
- Adriana Harakalova (C) 2019, 2024
- Jetpack Compose Logo
- Android Robot image https://developer.android.com/distribute/marketing-tools/brand-guidelines
- Kotlin logo https://resources.jetbrains.com/storage/products/kotlin/docs/kotlin_logos.zip
- Atomic Design https://bradfrost.com/blog/post/atomic-web-design/